Tampereen yliopisto

JANNE PAUSSU

DATAN VIRTUALISOINTI TERVEYSPALVELUISSA

Tekniikan ja luonnontieteiden tiedekunta

Diplomityö

Lokakuu 2019

TIIVISTELMÄ

Janne Paussu: Datan virtualisointi terveyspalveluissa
Diplomityö
Tampereen yliopisto
Johtamisen ja tietotekniikan DI-tutkinto-ohjelma
Lokakuu 2019
Tarkastajat: professori Tarmo Lipping ja yliopistonlehtori Jari Turunen

---

Tämän diplomityön päätavoitteet ovat selvittää mitä on datan virtualisointi, miten se toimii, onko datan virtualisoinnille tarvetta terveyspalveluissa ja miten datan virtualisointi soveltuisi terveyspalveluiden käyttöön. Aineistona toimivat alan kirjallisuus, markkinatutkimukset, ohjelmistotuottajien tiedotteet ja asiantuntijahaastattelut sekä tapaamiset. Menetelminä työssä on käytetty kirjallisuuskatsausta, markkinatutkimusten ja tiedotteiden selvittämistä, asiantuntijahaastatteluita ja lyhyttä ohjelmistotestausta. Ohjelmistotestauksen avulla selvitetään tekniikan käytännön puolen toimintaa ja soveltumista toimintaympäristöön.

Datan virtualisointi terminä yhdentyy kattotermiksi, jonka alla on useita tekniikoita. Yksittäin nämä tekniikat tarjoavat yksityiskohtaisia kohderatkaisuja. Datan virtualisoinnin tekninen puoli muodostuu näiden ratkaisujen toiminnasta ja kokonaiskuvasta suunnittelun lähtökohtana. Datan virtualisoinnissa yhdistyvät siis monet tekniikat, minkä lopputuloksena pyritään järjestelmään, jossa data tarjotaan käyttäjille yhden rajapinnan yli. Tämän rajapinnan yli data voidaan muokata ja tarjota käyttäjälle sellaisessa muodossa, että käyttäjän ei tarvitse erikseen tietää datan alkuperää tai tietokantarakennetta. Tiedon muokkauksessa tietoa voidaan jalostaa virtuaalitauluissa, jotka voidaan rakentaa verkostomaisesti tiedon päälle. Nykytekniikoista yleisimmät datan virtualisointia hyödyntävät ohjelmistot ovat datan federaatiota ja datan integraatiota suorittavat sovellukset. Nämä tuovat dataa useista lähteistä yhteen järjestelmään. Datan virtualisoinnin ratkaisuja on jo sovellettu dataputken toiminnassa, vaikka ei välttämättä juuri datan virtualisoinnin nimellä. Lyhyessä testauksessa havaittiin, että datan virtualisointi voi automatisoida ja yksinkertaistaa datan siirron toimintoja. Lopullinen kyvykkyys vaihtelee ohjelmistojen välillä.

Terveystiedon osalta datan monipuolisuus ja monilähteisyys asettavat haasteita, kun dataa integroidaan yhteisiin järjestelmiin. Näissä integrointimenetelmissä ja datan jalostuksen välivaiheissa datan virtualisointia voidaan käyttää helpottamaan työvaiheita. Datan virtualisoinnin periaatteita noudattavia menetelmiä onkin jo käytössä yksittäisissä kohdissa datanjalostusputkea. Datan virtualisointi ei kuitenkaan lopulta ole kokonaisvaltainen ihmeratkaisu, joka ratkaisisi kaikki ongelmat tai moninkertaistaisi työtehon kertaheitolla, mutta se sisältää hyviä menettelytapoja. Tulevaisuuden visioissa terveyspalvelut elävät muutoksen mukana, joka lisää terveyspalveluiden kysyntää ja samalla tarjoaa uusia mahdollisuuksia terveydenhoitoon. Näissä muutoksissa datan virtualisointi ja sen alle kuuluvat tekniikat ja menetelmät voivat olla hyödyksi.

Avainsanat: data, virtualisointi, virtualisaatio, terveys, terveydenhuolto, tieto

ABSTRACT

Janne Paussu: Data Virtualization in Healthcare Services
Master of Science Thesis
Tampere University
Master's Degree Programme in Management and Information Technology
October 2019
Examiners: Professor Tarmo Lipping and Senior Lecturer Jari Turunen

---

The main objectives for this thesis work were to find out what is data virtualization, how does it work, is there a need for it in healthcare data services and would it fit for such use. The material used were literature, marketing analyses, white papers from software producers and expert interviews along with meetings. The methods include studying the literature, studying the marketing analyses and white papers and using the interview materials along with a short software demo to find out how the practical side of things would fit operations.

Data virtualization as a term condenses into an umbrella term. Under the name 'data virtualization' are many techniques used for a specific local function and the technical side of the concept of data virtualization forms of using these solutions together as a basis for design. With these many techniques together, data virtualization strives for a single layer of API, i.e., application programming interface, where the data is integrated. Inside the system, the data can be transformed and sent towards the end user in a form such that the end user does not need to know the origin of the piece of data or the structural formation of the database the data is queried from. In the transformations of the data, virtual tables can be used. These virtual tables can be nested on top of each other creating a web-like structure. Data federation and data integration are the most common solutions utilizing data virtualization techniques. These bring data into a single system from multiple sources. Data virtualization solutions are already implemented in the use of data pipeline even though they are not named as data virtualization. In the short software demo, it was seen that data virtualization can automate and simplify data transfer functions. The capabilities of the software vary between vendors.

Part of the challenges for integrating healthcare data into unified systems come from the variety of sources and data types. In these integration-operations and midpoints in data refining pipeline data virtualization can be used to ease the stages and workload. Indeed, these principles of data virtualization are already in use in parts of the data refining pipeline. But in the end data virtualization is not some miracle solution for the entire pipeline that would solve all the problems and challenges and multiply the work efficiency in one fell swoop, yet it contains good practices. In the future healthcare must cope with the changes that will increase the demand for healthcare services and at the same time offer new capabilities and opportunities for patient care. In these probable future changes the techniques and practices of data virtualization can be useful.

Keywords: data, virtualization, health, healthcare, knowledge, information

ALKUSANAT

Diplomityön aiheen pariin johdattamisesta kiittäminen on toimeksiantoyhtiö 2M-IT:n Juhana Valoa ja Tampereen yliopiston Tarmo Lippingiä, jotka myöskin ovat kärsivällisesti toimineet ohjaajina työn tekemisessä. Diplomityö on ollut pidempi projekti kuin alun perin ajattelin. Matkan varrella olen oppinut paljon aihepiiristä ja toki omasta tutkimustyöstämisestä. Aihepiiriin liittyvä uutuudenkarheus on aiheuttanut turbulenssia tutkimustyössä ja lopullinen tavoite on tarkentunut ja jopa muuttunut matkan varrella. Tiedon löytäminen on ollut haasteellista ja terminologista tarkastelua on jouduttu tekemään. Ylämäet ja alamäet kuuluvat toki elämään. Lopultakin täydellistä selvitystä aiheesta tuskin pystyy yhden työn aikana antamaan.

Kiitokset työn tekemisessä auttamisesta 2M-IT:n väelle ja yliopistolla tarkastajina toimineille Tarmo Lippingille ja Jari Turuselle. Opinnoissa kiitoksen sana kuuluu koko Tampereen teknillisen yliopiston väelle ja toki aikaisemmissa opinnoissa koko opintoputken mahdollistaneelle kokonaisuudelle. Taustavoimina kaikessa tekemisessä on aina ollut perhe ja suku.

Työn varsinaisessa työstämisessä apuna ovat olleet ilmaislisenssin ohjelmat GIMP ja yEd unohtamatta muita työkaluja ja ohjelmistoja, jotka mahdollistivat työn tekemisen.

Porissa, 24.10.2019

Janne Paussu

SISÄLLYSLUETTELO

KUVALUETTELO

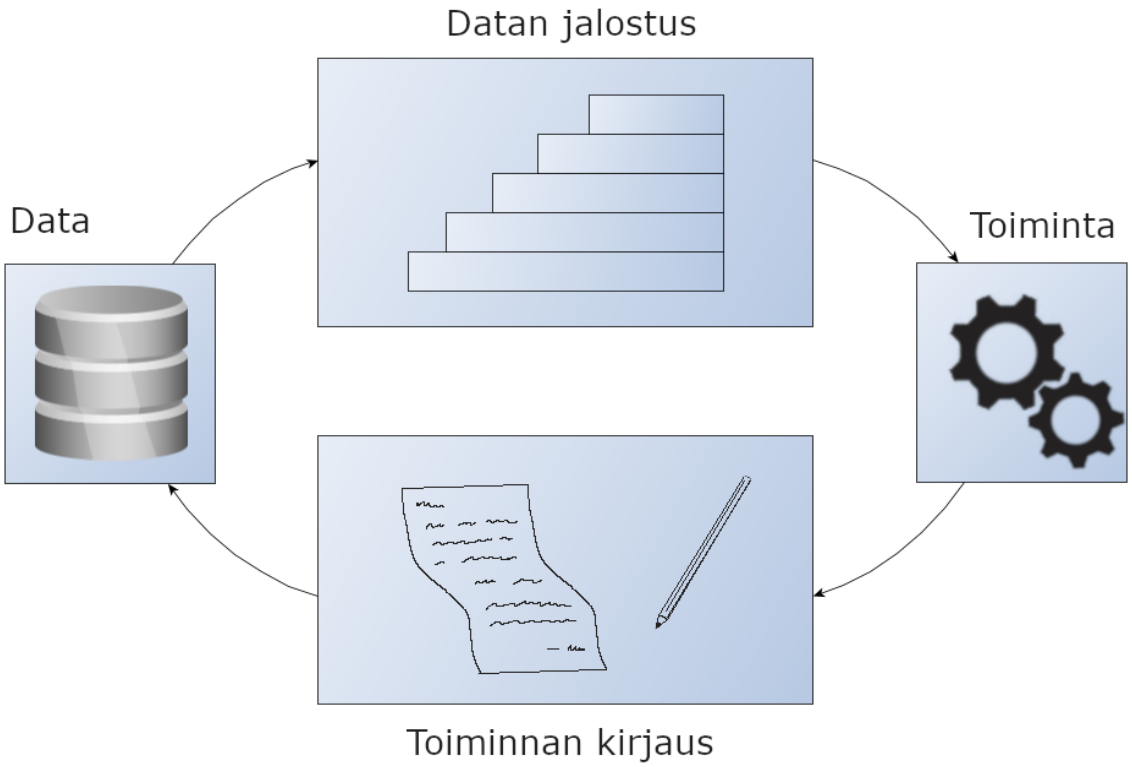
LYHENTEET JA MERKINNÄT

5G	Viidennen sukupolven langaton verkkoteknologia
API	Application Programming Interface, ohjelmointirajapinta
BI	Business Intelligence, liiketoimintatiedon hyödyntämistä
CD	Compact Disc, levymuotoinen tallennusmedia
ETL	Extract, Transform, Load, tiedon hakeminen, muuntaminen ja lataus
IoT	Internet of Things, esineiden internet
JDBC	Java Database Connectivity, ohjelmointirajapinta Java-kielelle
KBMS	Knowledge Base Management System, tiedonhallintajärjestelmä
OO-DBMS	Object-Oriented Database Management System, Oliopohjainen tiedonhallintajärjestelmä
SQL	Structured Query Language, kyselykieli
XML	Extensive Markup Language, merkintäkieli

1. JOHDANTO

Tallennettu data ja tieto ovat tärkeä tukipilari tietoyhteiskunnassa, jossa dataa tuottavia ja hyödyntäviä laitteita on kaikkialla, missä ihmiset toimivat. Tiedon soveltamisen merkitys läpäisee kaikki toiminnan tasot ja toimii oletusarvona toimintojen suunnittelussa, ohjaamisessa ja arvioimisessa. Tiedon korostunut rooli lisää sen tarvetta, ja tämän tarpeen tyydyttämisessä datan jalostus on merkittävässä osassa. Raakadatan saatavuus kasvaa sensoritekniikan ja tallennustekniikan sekä systeemien verkottumisen seurauksena, ja tämä data halutaan analysoida, jotta sen merkitys ja mahdollisuudet realisoituisivat. Vaikeasti löydettävästä ja siiloissa sijaitsevasta datasta ei saada täyttä potentiaalia eikä tämänkaltainen tieto saavuta sen tarvitsijoita. Terveydenhuollon alalla datan käytön ja hyödyntämisen mahdollinen potentiaali on suurta, sillä uutta sensoriteknologiaa kehitetään jatkuvasti ja samalla vanhat järjestelmät voivat olla vahvasti siiloutuneita tai sisältävät hyödyntämätöntä dataa valtavat määrät. Samanaikaisesti nykyhetken toiminta asettaa vahvoja vaatimuksia datan käytölle, jotta toiminnan tila ei häiriinny. Terveysdatan juuret ovat ihmisissä ja yksilöissä, jonka asettama vastuu voi olla hyvinkin elämän ja kuoleman kysymys. Potentiaaliset hyödyt datan jalostuksesta voivat siten myös koskea yksilöä, ihmistä, hyvin konkreettisella tavalla kun kyse on terveydestä. On siis perusteltua pyrkiä kohti tehokkaasti ja luotettavasti toimivaa dataa hyödyntävää järjestelmää.

Tämä diplomityö on tehty selvittämään datan virtualisoinnin käsitettä terveysdatan alalla ja se on tehty toimeksiantotutkimuksena 2M-IT -yhtiölle. Tavoitteena on selvittää tiedonjalostuksen toimintaa erityisesti terveysalalla sekä avata datan virtualisoinnin käsitettä. Aihetta ajavat eteenpäin ryhmä kysymyksiä, joihin työ pyrkii vastaamaan. Mitä on datan virtualisointi? Miten se toimii? Mitkä ovat datan virtualisoinnin kyvykkyydet? Voiko datan virtualisoinnilla vastata nykyhetken ja tulevaisuuden haasteisiin terveysdatan alalla? Menetelminä työssä toimivat kirjallisuuskatsaus ja markkinatilannekatsaus kyseisen ratkaisumallin eli datan virtualisoinnin osalta. Lisäksi mukana on pieni kokeilu käytännön ohjelmistosta, joka noudattaa datan virtualisointia. Aluksi selvitetään teoriaa datasta ja datan jalostuksesta sekä tiedonjalostusputken toiminnasta. Tämän jälkeen perehdytään käsitteeseen *datan virtualisointi* ja miten se sijoittuu datan jalostuksen aihepiiriin. Markkinakatsauksessa käytetään hyväksi markkinatutkimuksia selvitettäessä mitkä ratkaisut ja tarjoajat ovat datan virtualisoinnin tuottajia ja millaisia heidän datan virtualisoinnin ratkaisunsa ovat. Lopuksi käsitellään nykyhetken tilaa 2M-IT:n näkökulmasta dataputken suhteen asiantuntijahaastatteluiden avulla sekä selvitetään datan virtualisoinnin soveltuvuutta suhteessa nykyhetkeen ja tulevaisuuden tavoitteisiin.



Kuva 1. *Datan soveltaminen*

Kuvassa 1 näkyy diplomityön aihepiirin yleinen ympäristö. Data itsessään on ”kuollutta”, mutta sitä jalostamalla voidaan päästä toimintaan, jolla on suora vaikutus maailmaan. Toiminta tässä tapauksessa merkitsee ihmislähtöistä suoritusta, jolla on jokin tavoite. Toiminta taas usein tuottaa uutta dataa. Vaikka kuvassa tämä kulku on syklinä, se ei sitä välttämättä ole. Kuvan tärkein asia on havainnollistaa yhteys ja toisaalta ero datan ja toiminnan välillä ja sijoittaa diplomityö näiden välille datan jalostuksen raameihin.

Datan virtualisointi on terminä vielä uudenkarhea eikä näin ollen hyvin rajattu. Aiheesta löytyy myös niukasti materiaalia. Tästä syystä aiheen asiantuntijan Rick van der Lansin kirjoitukset ja kirja ovat isossa roolissa datan virtualisoinnin määrittelemisessä. Rick van der Lans on julkaissut kirjan ”Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses” (van der Lans, 2012), joka käsittelee dataa liiketoimintatiedon alan näkökulmasta, mutta samat periaatteet pätevät myös muuhun dataan. Terveysdatan ja terveydenhoitoalan tulevaisuuden trendeissä on isossa roolissa kirja ”Health 4.0: How Virtualization and Big Data are Revolutionizing Healthcare” (Thuemmler, 2017).

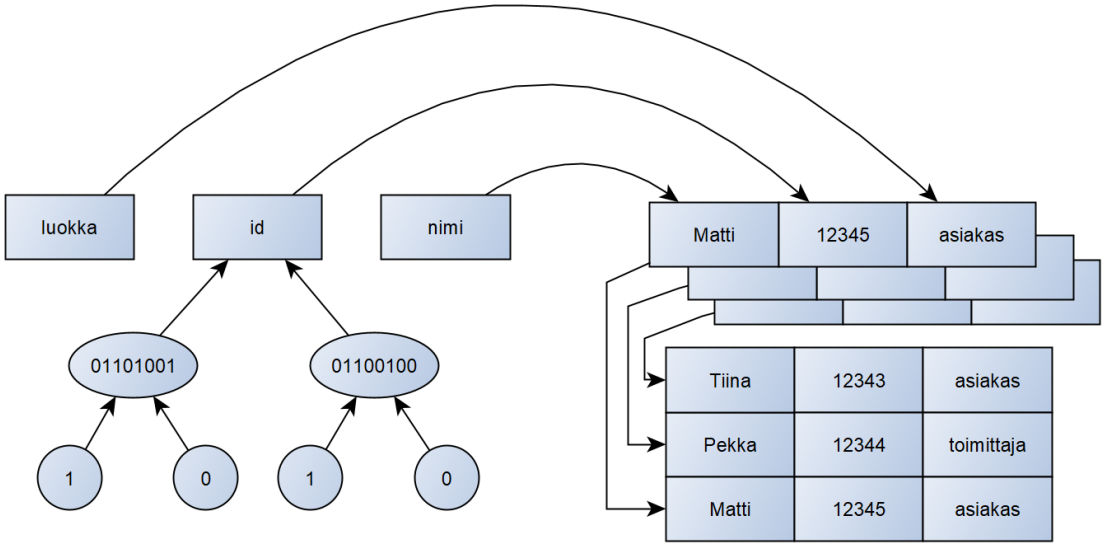
2. DATAN JALOSTAMISEN TAUSTA JA TEORIA

Teoriana diplomityön tekemisessä toimivat tietokantaratkaisujen ja tiedonsiirron ja -käsittelyn periaatteet. Tässä kappaleessa käsitellään aihepiiriin liittyvää teoriaa ja esitetään keskeisimmät käsitteet, joita työn tekemisessä tarvittiin.

2.1 Perinteisen datan käytön ja tietokantasovellusten kehitys

Mitä on data ja mitä puolestaan tieto? Suomeksi tiedon eri asteille ei ole yksiselitteisiä nimiä, mutta niitä voidaan kuvailla olevan data, informaatio ja tieto. Tässä järjestyksessä tiedon jalostusaste kasvaa. Joissakin lähteissä tiedolle kuvaillaan vielä ylempi jalostusaste, viisaus, joka on muodostettu tiedosta induktiivisesti ja jonka avulla uutta tietoa voidaan käsitellä deduktiivisesti, mutta sitten mennäänkin jo filosofian puolelle (Jancsurak, 2013). Dataa voidaan pitää siis tiedon ensimmäisenä asteena, mutta data itsessään ei välttämättä ole vielä tietoa. Datasta voidaan kuitenkin jalostaa tietoa, joka on loppukäyttäjälle hyödyllistä (McFadden, Hoffer ja Prescott, 1999). Seuraava esimerkki kertoo datan ja tiedon suhteesta tietokannoissa. Perinteisiä tietokantojen hyödyntäjiä ovat olleet esimerkiksi lentoyhtiöt ja muut toimijat, jotka joutuvat pitämään kirjaa monista varauksista monissa eri paikoissa. Lisäesimerkkeinä toimivat pankit tileineen ja yritykset liiketoimintatiedon kanssa (Ullman ja Widom, 1997). Tämänkaltainen rekisteri koostuu datasta, joka ei vielä itsessään kerro mitään muuta kuin, että jotkin tapahtumat tai faktat on talletettu ja kirjattu talteen. Näistä tiedon murusista ei vielä näe kokonaisuutta, mutta niistä voidaan hakea ja yhdistellä mielekäs tieto, joka voidaan liittää haluttuun asiayhteyteen, esimerkiksi hakea tietyn asiakkaan tiedot ja tilaukset. Tähän datan ja tiedon jalostamiseen tarvitaan tietokantasovelluksia ja -systeemejä, joiden avulla tietokannoissa talletettu data yhdistetään tiedoksi.

Atomisen soveltamattoman datankin sisällä on hierarkiaa, joka määrittää datan olemusta. Pohjalla tietokoneavusteisesti luettavassa datassa on binäärinen bitti, joka on joko 1 tai 0. Bittijonoista koostuvat tavut. Yksi tavu voi määrittää esimerkiksi yhtä merkkiä, kuten kirjainta tai numeroa. Tavujen määrittämistä merkkijonoista koostuvat sanat tai luvut, jotka voivat määrittää tietokannassa kentän (*field*) tai sarakkeen. Sanojen ja lukujen yhdistelmä voi muodostaa yksilöidyn tietoalkion (*record)*. Tietoalkioiden joukko muodostaa tiedoston tai kansion (Buchholz, 1962). Tätä suhdetta kuvataan Kuvassa 2.



Kuva 2. *Datan hierarkian kuvausta Buchholz:n taulukkoa mukaillen* (Buchholz, 1962)

Kuvassa 2 vasemmalla alhaalla kuvataan binääriset bitit 1 ja 0, jotka muodostavat tavuja, jotka muodostavat sanan tai numeron. Kuvassa 2 nämä sanat tai numerot ovat sarakkeiden otsikoita, vaikka sisällöllisesti ne ovat tietoalkion attribuuttien arvoja, kuten nähdään Kuvan 2 oikeassa reunassa. Nämä sanat muodostavat yksittäisen rivin, joka on yksilöity entiteetti. Näistä yksilöistä koostuu tiedosto, joka on siis kokoelma rivejä.

Tietokantojen historia limittyy tietokonetekniikan kehittymiseen, kun puhutaan koneavusteisesta tiedonkäsittelystä. Tietoa ja arkistoja on pidetty fyysisessä tallessa esimerkiksi paperisena jo historian alkuajoista saakka. Tässä työssä keskitytään kuitenkin tietokoneavusteiseen tietokantojen käsittelyyn ja dataan, joka taas on verraten tuore soveltamisala. Jo 1970-luvulla on tunnistettu, että loppukäyttäjää ei kuitenkaan yleensä kiinnosta itse datan jalostumisen prosessi ja käsittely, sillä tiedon loppukäyttäjä ei useinkaan ole tiedonjalostuksen ammattilainen. Samalla tietokantasysteemit ovat perinteisesti olleet manuaalisesti hyvin raskaita rakentaa, jolloin tiedonjalostuksen ja käsittelyn osaamista on tarvittu, vaikkakin automaattisista ja systemaattisista apuvälineistä on toivottu ratkaisua pitkään (Sundgren, 1975).

Tietokantajärjestelmissä pohjalla toimiva tietokanta on usein toiminut relaatiotietokantamallin mukaan, mutta näin ei aina ole ollut, vaan hierarkkiset ja verkkomallit ovat olleet ennen yleisiä, osittain tietokonetekniikan rajoitteiden vuoksi. Relaatiotietokantojen teorian alkuvaiheessakin kiinnostus relaatiotietokantoihin oli kuitenkin vahvaa, vaikka käytännön sovelluksia ei vielä ollut (Date, 1977). Sittemmin relaatiotietomallin voittokulkua kuvaa relaatiomallin pitäminen koko tietokanta-alan perustana. E. F. Coddin relaatiotietomallin teorian julkaisemisen jälkeen (Codd, 1969), relaatiomalli on laajentanut teoreettista pohjaa itse datan ja sen suhteiden käsittelemiseen (Date, 1995). Muutosta on kuitenkin tapahtunut lähes joka vuosikymmenellä ja tietokantojen systeemitekniikan suuntaus

on vaihdellut. Tietokantamallien välisiä eroja on arvioitu alla olevassa Taulukossa 1. (Ullman, 1988).

Taulukko 1. *Tietokantasysteemien historian vallitsevien teorioiden eroja Ullmanin taulukon mukaan* (Ullman, 1988).

Vuosikymmen	Systeemi	Orientaatio	Deklaratiivinen	DML (Data Manipulation Language), datan muokkauskieli
1960	Verkkomainen, hierarkkinen	Objekti	Ei	Erillinen
1970	Relationaalinen	Arvo	Kyllä	Erillinen
1980	OO-DBMS, oliopohjainen	Olio	Ei	Integroitu
1990	KBMS, Knowledge Base	Arvo	Kyllä	Integroitu

Taulukko 1 on julkaistu vuonna 1988, jolloin 90-luvun osalta on käytetty arviota tulevasta. Sittemmin Knowledge Base -pohjaiset järjestelmät kehittyivät asiantuntijajärjestelmien (*expert system*) suuntaan. KBMS pohjautuu suureen aineistopohjaan ja tiedon hakemiseen loogisin säännöin, jolloin haetaan mahdollista ratkaisua kysymykseen, joka voi olla monitahoinen ja vaatia sääntösuhteiden tulkintaa. Merkittävä tekijä sääntösuhteissa on ontologia, joka määrittelee käsitteiden suhdetta toisiinsa. Käsitteet tai termit voivat olla sanoina erilaisia, mutta tarkoittavat silti samaa asiaa tai niillä voi olla jokin muunlainen suhde. Esimerkiksi asiakaspalautteiden kuvauksissa voi olla tarpeen hakea kaikki kielteisiä sanoja sisältävät, jotta voidaan selvittää mahdollisia ongelmia. Tällöin tarvitaan ontologisia termien suhteita, jotta voidaan selvittää mitkä ovat kielteisiä termejä. Tämä on erilainen hakutoimenpide kuin yksittäisen termin haku. Asiantuntijajärjestelmä eroaa KBMS:stä siten, että asiantuntijajärjestelmän sisältö on kuratoitu vastaamaan kokoelmaa tietoa kyseiseltä erikoisalalta.

1990-luvun aikana ja sen jälkeen internet on kokenut leviämisen nousukauden ja vakiinnuttanut asemansa jokapäiväisessä toiminnassa. Tämän laajan verkostoitumisen mahdollisuudet muuttivat osittain myös tietokantojen fokuspistettä kehityksessä kohti kyvyk-

kyyksiä toimia eri rajapintojen yli. Tämä johtuu siitä, että internetin mahdollistama yhteyden luominen verkon yli avasi mahdollisuuden laajentaa yhteydenpitoa eri järjestelmien välillä. Yhdistettävien järjestelmien ei enää tarvinnut sijaita samassa paikassa fyysisesti tai samassa sisäverkossa.

Tiedon tallentamisessa suuri määrä dataa on ollut perinteinen haaste, jolloin tietoa on saattanut joutua tallentamaan hitaammille formaateille. Nämä ovat olleet käytännössä levyratkaisuja, joiden dataa on luettu mekaanisesti tai hajautetulle datalle on jopa fyysisesti haettu tarvittava tallennusmedia kuten CD. 1990-luvun lopulla muun muassa multimedian määrä tallennettavasta datasta alkoi kasvaa, joka asetti omat haasteensa suurien datamäärien tallentamiselle. Erityppistä signaalidataa saatettiin joutua tallentamaan gigatavujen kokoisina yksikköinä, jolloin niiden käsittely oli kömpelöä tietokantasysteemeille ja tietokonejärjestelmillä niiden siirtäminen voi vaatia lisätyövaiheita. Internetin mahdollistamana eri paikoissa sijaitsevia tietokantoja voitiin integroida helpommin, jolloin törmättiin seuraavaan haasteeseen, joka on erityyppiset tietokantatyypit. Nämä eri tietokannat voivat kattaa monenlaisia järjestelmiä ja tietotyyppejä. Datan integraation seurauksena syntyi tietovarasto (*data warehouse*) johon talletetaan kaikkien integroitavien tietokantojen tiedot yleensä säännöllisin välein kuten päivittäin. Tämän tietomäärän keskittyminen toi myös uuden käsitteen tiedon louhinta (*data mining*) jolla saatavat edut olivat vielä 90-luvulla lähinnä teorioita löydettävistä suhteista ja kuvioita tiedosta kuten metatietoa (Ullman ja Widom, 1997). Myös tekstidatan tallennuksessa on tehty töitä, jotta dokumenteista voidaan saada talteen sekä rakenne että sisältö. Tällöin käytetään usein jotakin rakenteellista kieltä kuten esimerkiksi XML:ää (Chin, 2001).

Myös kyselykieliä on useita, joista tunnetuin lienee SQL, joka on saanut aikanaan osakseen kritiikkiä muun muassa redundanttisuudesta. Tämä voi lisätä käyttäjäystävällisyyttä, sillä ratkaisuja voi olla monia. Samalla lisätään kuitenkin vastuuta valita kyselyyn tehokas kyselylause, koska eri kyselylauseet voivat olla keskenään erinopeuksisia (Desai, 1990). Käyttäjälle voidaan tarjota myös hakua helpottava sovellus, joka suorittaa haun fyysisen puolen, kunhan käyttäjä antaa sille hakutermejä. Nämä hakukoneet liittyvät kuitenkin enemmän sellaisen tiedon hakemiseen, jossa käyttäjä ei aina tarkalleen välttämättä edes tiedä mitä on hakemassa. Tämänkaltaiset hakukoneet voivat käyttää luonnollisen kielen synonyymeja löytääkseen asiaan liittyviä osumia, toisin kuin liiketoimintatietokannoissa, joista käyttäjä hakee hyvin spesifisen tiedon, joka liittyy juuri tiettyihin avaimiin muodostaakseen niistä raportin (Chin, 2001). Tämä hakukoneiden suorittama semanttinen haku on kehityksellisesti jatkumoa aikaisemmin mainituille asiantuntijajärjestelmien toiminnalle.

Tietokantasysteemin rakenteen suunnittelussa on havahduttu jo 80-luvulla suunnittelun tärkeyteen sekä loppukäyttäjien tarpeiden selvittämisessä että tehokkuuden riittävyydessä. Suunnittelussa on otettava huomioon koko organisaatio (Teorey and Fry, 1982).

Ratkaisujen vaatimukset on hyvä muotoilla ensin ja valita työkalut sen mukaan. Jos toimitaan toisinpäin, on riski, että työkalu ei sovellu käyttötarkoitukseen (McFadden, Hoffer ja Prescott, 1999).

Liiketoimintatiedon hyödyntämisessä törmätään nopeasti haasteisiin, joihin on olemassa perinteisiä ratkaisuja. Tiedon rakenteessa ja sijainnissa voi olla rajoitteita, jotka estävät tiedonkäyttäjiä saamasta tarvitsemaansa tietoa siinä muodossa kuin sen tarvitsevat tai sillä hetkellä, kun sitä vaaditaan. Tähän voivat vaikuttaa muun muassa tiedon sijainti tallennusvälineellä, joka ei kestä kuormitusta monelta tiedonlukijalta. Lähdetietokanta on useimmiten myös operatiivinen tietokanta, johon tietoa tallennetaan aktiivisesti. Tällöin useiden kyselyjen aiheuttama kuorma ja siitä mahdollisesti seuraava hidastuminen ei haittaa ainoastaan raportin lukijoita, vaan se voi haitata jopa itse toimintaa, jonka tarkoitus on tallentaa tiedot lähdekantaan (Linstedt *ym.*, 2016).

2.2 Nykyhetken viitekehys ja big data

Perinteisen tiedonkäsittelyn kehityksen seurauksena on monia nousevia trendejä. Koska digitalisaatio on levinnyt kaikkialle, on datan määrä kasvanut eksponentiaalisesti. Samalla suorituskyky ja yhteydet järjestelmien välillä ovat kehittyneet. Yhdessä nämä seikat luovat uusia mahdollisuuksia, jotka tuovat mukanaan myös ongelmia. Koska mahdollisuudet datan hyödyntämiselle ovat muuttuneet, se muuttaa myös vaatimuksia ja odotuksia. Business Intelligence-alalla data on hyödynnettävä tai kilpailijat kuittaavat sen potentiaalin. Kapasiteetin kasvu lisää tiedon tallentamista, mutta jos dataa ei hyödynnetä joustavasti, se uhkaa siiloutua ja datasta mahdollisesti saatava etu jää toteutumatta. Big datan yhteydessä usein palveluntarjoajat käyttävät käsitettä 4V, *Volume, Variety, Velocity* ja *Veracity*. Näissä käsitteissä *volume* merkitsee volyymia tai määrää, *variety* moninaisuutta tai erityyppistä dataa, *velocity* tietovirran analysointia ja *veracity* tiedon varmuutta ja luotettavuutta (*IBM Big Data Analytics Hub*, 2018). Nämä käsitteet kuvaavat big datan mahdollisuuksia ja haasteita. Kukin näistä ominaisuuksista mahdollistaa uusia toimintatapoja ja samalla niiden hyödyntäminen vaatii muutoksia. Terveysdatassa näihin käsitteisiin voidaan lisätä kaksi V:tä, *Value* eli tiedon arvo ja *Variability* eli tiedon kausiluontoisuus (Andreu-Perez *ym.*, 2015). Tiedon arvolla voidaan tarkoittaa esimerkiksi diagnoosin saamista aikaisemmin, joka on potilaalle ja hoitohenkilökunnalle arvokasta tietoa. Tiedon kausiluontoisuudella voidaan tarkoittaa mm. epidemioita ja niiden ennustamista ja mallintamista.

Sensoreiden ja yhteyksien kehittyminen ja suorituskyvyn kasvaminen sekä datan hyödyntämisen realisoituminen ovat luoneet käsitteen Industry 4.0 teollisuuden alalle ja siihen liittyvät esimerkiksi esineiden internet (*Internet of Things, IoT*), ja pilvipalveluiden hyödyntäminen. Tätä periaatetta voidaan käsitellä myös muilla aloilla. Terveysalalla sama käsite on kuvattu nimellä Health 4.0. Tietoverkkojen kehityksessä 5G-teknologioilla luvataan olevan paljon potentiaalia. Viiveiden pienentyminen ja kytkettävien laitteiden määrän kasvu yhdessä verkon kattavuuden, akkukestävyyden ja palvelutasojen nousun

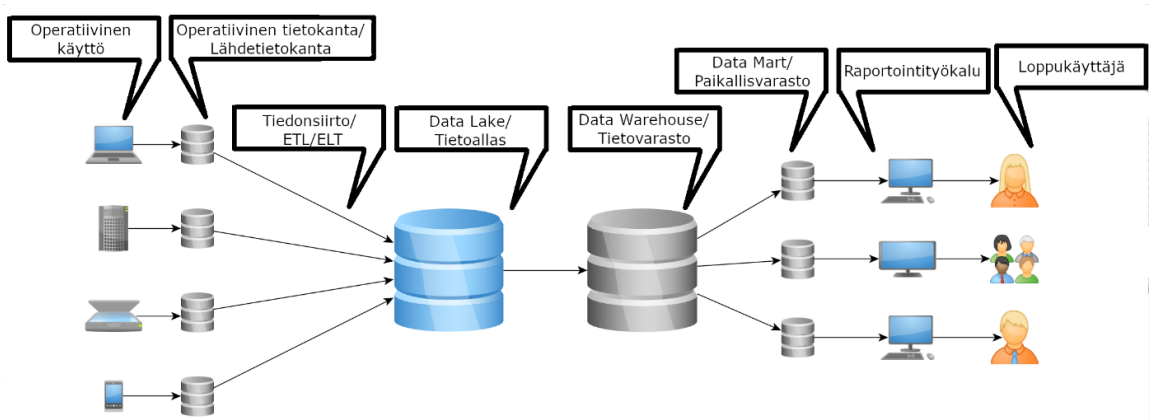
kanssa tuovat IoT:n myös terveydenhuoltoon. Uusien laitteiden ja niiden ohjaamisen seurauksena voidaan olettaa, että kerättävää dataa kertyy myös lisää. Tämä data täytyy myös käsitellä ja säilöä asianmukaisesti, mikä luo haasteita tiedonjalostukseen, jos kapasiteetin tarpeen kasvu on vahvaa. Mahdolliset hyödyt datan määrän kasvusta ovat moninaisia. Esimerkiksi riippuvuussuhteille voidaan kehitellä laskukaavoja, jos saadaan enemmän dataa tilanteeseen vaikuttavista tekijöistä. Terveyden osalta tämä voi olla tutkimusta esimerkiksi jonkin mitatun tekijän vaikutuksista sairauden etenemiseen. Samalla kun datan määrän kasvaminen volyymina ja lähteiden määrässä on visioitu kasvavan, datan kulkeminen ja liikkuminen eri tahojen välillä on haastava arvioitava eritoten terveysdatan saralla. Vaikka dataa kertyisikin enemmän, dataa ohjaavat monet erityiset säädökset sen sisältäessä yksilölle henkilökohtaisia asioita. Tästä syystä ei voida olettaa, että lisääntynyt terveysdata kulkisi esimerkiksi tutkimuskäyttöön. Jo operatiiviset toimijat kuten eri alueiden terveyskeskukset eivät aina voi vaihtaa keskenään potilaasta terveysdataa, vaikka potilas olisi asiakas molemmissa. Tämä vaatii aina potilaan luvan ja järjestelmien yhteensopivuuden. Terveydenhuollon ala on elänyt kehitystä, jossa potilaan läpimenoaika pyritään minimoimaan. Tällöin potilaalle kertyy vähemmän sairaalapäiviä ja sairaslomapäiviä. (Thuemmler, 2017).

2.3 Yksinkertainen esimerkki tiedonjalostusputkesta

On olemassa monia tiedon säilytysratkaisuja ja ohjelmia, joilla tietoa säilötään, siirretään ja haetaan. Yksinkertaisin ratkaisu tiedon jalostamisessa olisi, jos loppukäyttäjä kävisi hakemassa tarvitsemansa tiedon suoraan lähteestä. Tämä edellyttää, että käyttäjä tietää, mitä tietoa hän haluaa, missä se sijaitsee, miten se haetaan ja missä muodossa se on. Tämän ei tarvitse olla monimutkainen prosessi; esimerkkinä voidaan ajatella yksinkertaista laskuria, jossa on juokseva luku vaikkapa asiakkaista, jotka ohittavat laskurin. Käyttäjä voi käydä vaikka fyysisesti paikan päällä lukemassa asiakkaiden määrän, jonka laskuri on tallentanut ja muistaa verrata sitä mielessään tai paperilla edelliseen lukemaan. Tämä skenaario ei tiedonhallintajärjestelmissä ole realistinen, mutta voi toimia esimerkkinä tiedon jalostusputken rakenteen muodostumiselle. Jos esimerkkiin lisätään uusia laskureita, työn määrä lisääntyy, kun jokaisen laskurin luona pitää käydä erikseen. Laskurin tarkastajia voidaan myös lisätä, jolloin saavutaan moni moneen suhteeseen, jolloin havaitaan nopeasti, ettei ole mielekästä, jos jokainen käyttäjä joutuisi käymään jokaisella laskurilla erikseen. Jos oletetaan, että kaikki laskurit ovat fyysisesti samassa paikassa ja samanlaisia keskenään, voisi yksi ”muistaja” käydä lukemassa kaikki laskurit ja käyttäjät voisivat käydä kysymässä tältä muistajalta keskitetysti laskurien tiedot. Entä jos laskureita on eri sijainneissa ja erityyppisiä laskureita, joita muistaja ei osaa käyttää? Mukaan voidaan lisätä ”kerääjä”, joka käy hakemassa laskurien tiedot muualta ja toimittaa ne muistajalle. Lisätään vielä käyttäjiä, jolloin käyttäjät joutuisivat jonottamaan vuoroaan muistajalle. Nyt voidaan lisätä ”jakelijoita”, joille muistaja antaa tiedot laskureista ja käyttäjät voivat

kysyä omalta jakelijaltaan tiedot laskureista jonottamatta. Tässä yksinkertaistetussa esimerkissä tieto kulkee fyysisesti henkilöiden mukana, ja varsinaisissa tietojärjestelmissä on myös huomioitava tiedon kulkeminen eri järjestelmien välillä.

Esimerkistä voidaan poimia lähdekannaksi laskuri ja loppukäyttäjäksi laskurin luvun tarvitseva käyttäjä. Kerääjä esittää *data lake* -ratkaisua ja muistaja *data warehouse* – ratkaisua eli tietovarastoa. Jakelijat ovat *data mart:*eja. Tiedon siirtäminen toiselle toimijalle on ETL tai ELT -prosessi joiden kirjaimet tulevat sanoista *Extract, Transform* ja *Load* eli haku, muokkaus ja lataus. Tarkempia esittelyjä ja perusteluja käytölle näistä kyseisistä menetelmistä ja tekniikoista on seuraavissa kappaleissa. Kuvailtua työnjakoa on havainnollistettu Kuvassa 3, joka on yleinen muoto tiedonjalostusputkelle.



Kuva 3. *Tiedonjalostusputken yleinen muoto ja osatekijät*

Kuvassa 3 on tiedonjalostusputki kuvattu sen yleisessä muodossa. Kuvassa data kulkee vasemmalta oikealle nuolien kuvaamassa järjestyksessä. Vasemmalla on dataa tuottava prosessi, jonka data tallennetaan operatiiviseen tietokantaan. Tästä tietokannasta data pitää siirtää mahdollisesti tietoaltaaseen tai tietovarastoon tai muuhun kuormitusta kestävään ympäristöön. Tietoallas toimii kokoajana ja jäsentelijänä ennen datan siirtämistä tietovarastoon. Tietovarastosta dataa ohjataan paikallisvarastoihin siinä muodossa ja siinä määrin kuin sitä tarvitaan seuraavassa askeleessa. Paikallisvarastosta dataa haetaan raportointityökaluilla, jotka muodostavat raportin loppukäyttäjille.

2.4 Tiedonsiirtoprosessi

Ennen kuin tietojärjestelmää voidaan käyttää, tarvitsee se tietenkin tietoa. Operatiivisessa kannassa tieto kerääntyy usein juoksevasti jatkuvalla syötöllä hippunen kerrallaan. Tällöin tärkeintä on, että operatiivinen tietokanta pysyy operatiivisena eli toimivana ja sinne voidaan syöttää tietoa tarvittaessa. Jos tiedot ajettaisiin eteenpäin jatkuvana syöttönä voisi se rasittaa operatiivista lähdekantaa niin, että sen toiminta voisi häiriintyä. Siksi yleinen malli on ajaa lähdekannan tiedot eteenpäin yhtenä eränä määräajoin, usein silloin kun lähdekanta on vähemmällä operatiivisella käytöllä, esimerkiksi öisin.

Yhden erän ajo tiedonsiirrossa voi olla hyvinkin massiivinen urakka, jos kyseessä on koko tietoaineiston kattava pullonkaula tietoputkessa. Silloin on merkitystä, milloin, miten ja missä järjestyksessä erän siirto tapahtuu. Yleinen lyhenne tiedonsiirron yhteydessä on ETL tai ELT. Haku tapahtuu luonnollisesti lähettävän kannan puolella, jolloin se käyttää resursseja hakeakseen tietoerään kuuluvan datan. Latauksessa tiedon vastaanottava kanta käyttää resursseja siihen, että eräajon tieto talletetaan sinne. Välissä on kuitenkin *data staging* -alue, jonne tieto haetaan ja josta se ladataan seuraavaan kantaan. Tässä välialueella voidaan suorittaa muokkaus, jossa tietoa muokataan sopivaan muotoon seuraavaa kantaa varten ja sille tehdään tarkastuksia. Perinteisessä järjestyksessä (ETL) tiedolle tehdään määritetyt muokkaukset ja tarkistukset, jonka jälkeen ne ladataan vastaanottavaan kantaan. Muokkauksen ollessa päällä tiedot eivät ole käytettävissä. Uudemmassa ELT-mallissa tiedot ladataan ensin esimerkiksi pilvipalvelujen mahdollistamana yhtenä eränä, jonka jälkeen niille voidaan suorittaa muokkauksia ja tallentaa ne seuraavaan kantaan. Tässä ELT-mallissa tietoja voidaan tarkastella ennen niiden muokkausta ja tallentamista, jolloin muokkauksen parametrejakin on mahdollista vaihtaa. Suorituskyvystä riippuen voi olla hyötyä, jos *transform*-vaihe voidaan siirtää vastaanottavan kannan päätyyn, varsinkin jos vastaanottava kanta kykenee siihen.

2.5 Data lake

Data lake eli tietoallas kerää useista eri lähteistä tietoja yleensä sellaisenaan. Tämä siirto voidaan suorittaa ELT-prosessilla, jolloin datasetti on siirretty tietoaltaaseen sellaisenaan, ja sitä voidaan muokata vasta silloin kun sitä tarvitaan, olettaen, että datan mukana on talletettu metatiedot lähdekannasta, jotta tiedon alkuperä pysyy selvillä. Tämä ELT-pohjainen ratkaisu metatietojen kera mahdollistaa nopean siirron tietoaltaaseen ja toisaalta tietoaltaasta eteenpäin. Kun tieto on kerätty tietoaltaaseen, voidaan sitä helpommin hakea sieltä vaikkapa tietovaraston tarpeisiin. Kun tieto siirtyy tietovarastoon, tehdään viimeistään tässä vaiheessa tietovaraston vaatimat toimenpiteet datan muokkauksella, jotta data vastaa tietovaraston vaatimuksiin. Tietoaltaan ketteryys mahdollistaa sen toiminnan kerääjänä toimijana, joka kykenee hakemaan monenlaista dataa monenlaisista kohteista, vaikka kyseinen data olisi rakenteetonta, rakenteellista tai puolirakenteetonta. Tietovarasto ei välttämättä taivu niin helposti näihin vaatimuksiin. Kun monet eri tietolähteet saadaan yhden käyttöjärjestelmän alle, voidaan ehkäistä senkaltaista tiedon siiloutumista, jossa eri käyttöjärjestelmien käyttäjillä on kynnys lähteä hakemaan dataa mahdollisesti vieraasta käyttöjärjestelmästä. Tietoaltaan ketteryyden mahdollistaa usein avoimen lähdekoodin ja varsinkin pilviratkaisuiden muodostama laajeneva kokonaisuus, jossa lisää kapasiteettia otetaan käyttöön tarvittaessa. Tämä onnistuu, koska tietoallas ei aseta niin tiukkoja vaatimuksia tiedolle ja tietokannan rakenteelle kuin esimerkiksi tietovarasto (Khine ja Wang, 2018). Data tietoaltaassa voi olla luonteeltaan hyvin raakaa, jolloin siitä ei aina suoraan voida tehdä välttämättä analyyseja, mutta se mahdollistaa tiedon saatavuuden jatkojalostusta varten.

2.6 Data warehouse

Data warehouse eli tietovarasto antaa mahdollisen ratkaisun moneen pulmaan. Tietovarastossa tieto yhdistetään mahdollisesti useista eri järjestelmistä yhteen tietokantaan varastoon. Tämänkaltainen toimenpide suoritetaan yleensä ETL- tai ELT -prosesseina, jotka esiteltiin tiedonsiirtoprosessin yhteydessä.

Jos tietolähteitä on monia ja niiden järjestelmät ovat keskenään erilaisia, tietojen koostaminen voi olla vaikeaa. Jos analyysia varten raporttiin tulee hakea tietoa eri tietokannoista ilman tietovarastoa, tulee raportoinnin rakentamisesta työlästä, kun huomioidaan jokaisen tietokannan vaatimukset tiedon rakenteessa ja tiedonsiirron yhteydessä. Voi myös olla, että seuraava raportti vaatii jälleen erilaisten lähdekantojen huomioonottamista, jolloin työn määrä kertaantuu joka raportille. Tietovaraston hyödyntäminen tarjoaa tässä tapauksessa yhden yhtenäisen näkymän tiedoista, jotka on sille talletettu. Kun tietoa kysellään ylempää tiedonsiirtokanavalta, voidaan toimia yhden järjestelmän puitteissa, eli tietovaraston, jolloin raportoinnissa ei tarvitse enää tehdä lisätyötä alkuperäisten lähdekantojen kanssa toimimiseen (Hovi, 1997). Tietovarasto kuitenkin usein asettaa vaatimuksia tiedon rakenteelle. Siksi tietovaraston ja lähdekantojen välillä voi olla toiminnassa tietoallas.

Alkuperäisten operatiivisten lähdekantojen kuormittaminen raportointikyselyillä voi siis haitata itse operatiivista toimintaa. Jos tiedot ajetaan määräajoin hallitusti tietovarastoon, voidaan kuormitus operatiivisiin kantoihin yleensä ennakoida ja sovittaa hallitusti sopivaan ajankohtaan. Vaihtoehtoisesti operatiiviset tietokannat voivat lähettää tietonsa tietoaltaaseen, josta ne voidaan siirtää tietovarastoon. Raportointityökalut voivat näin ollen kuormittaa tietovarastoa kyselyillä, mikä ei haittaa operatiivisia lähdekantoja.

Tietovarastoon tiedot voidaan ajaa yhdellä kyselyllä lähdekannoista tai tietoaltaasta esimerkiksi kerran päivässä. Tietovarasto tallentaa jokaisen eräajon tuloksena syntyneen tietokattauksen. Tämä mahdollistaa historiatiedon tallentamisen ja vertailun, kun voidaan hakea esimerkiksi viime viikon tilanne ja verrata sitä nykyhetkeen. Operatiivisissa lähdekannoissa usein tieto on esimerkiksi juoksevana numerona, jolloin historiatietoa ei ole. Koska lähdejärjestelmien tieto tallentuu määräajoin tietovarastoon aikaleimattuna tilannekattauksena, mahdollistaa se tietovaraston toimimisen yhtenäisesti eheänä tietolähteenä (*Single Source of Truth*, SSOT). Tiedonsiirron yhteydessä yleensä myös tiedon rakenne jalostuu, jos sille tehdään muokkauksia, jotta se vastaa tietovaraston vaatimuksiin. Tämän jälkeen tieto voi olla jo analysoitavissa tai tulkittavissa raportteihin.

2.7 Data mart

Data mart on seuraava askel tietovaraston jälkeen tiedon tarjoamisessa eteenpäin. Yksi mahdollinen nimitys *data mart:*ille voisi olla paikallisvarasto, koska sen sisältämä data on usein paikallisen tarpeen mukainen. Tieto voitaisiin hakea myös ilman *data mart*:ia suoraan tietovarastosta, ja joissakin tapauksissa tämä olisi täysin toimiva ja ongelmaton

tapa toimia. On kuitenkin tilanteita, joissa ongelmia voi syntyä. Kun tietovaraston sisältämän tiedon käyttäjien määrä kasvaa, kasvaa myös hakujen määrä ja rasitus tietovarastoa kohtaan. Kun tieto ohjataan *data mart*:eille, käyttäjät voivat hakea tarvitsemansa tiedot niistä ja rasittaa niitä ilman, että tämä rasitus näkyisi taaksepäin tiedonjalostusketjussa. Tietovaraston tiedot ovat usein atomisessa muodossa ja ei ole useinkaan mielekästä tallentaa samaa tietoa muokattuna kopiona. Tämä loisi redundanttisuutta, jolloin tietovarasto ei toimisi *single source of truth* -pisteenä. Käyttäjillä voi kuitenkin olla tarve muokata ja yhdistellä tietoja omiin tarpeisiinsa. Kun tiedot ovat *data mart*:issa, ne voidaan räätälöidä loppukäyttäjille sopivaan muotoon ja yhdistellä tarvittavien datojen kanssa. Vaikka data on eri muodossa tai yhdisteltynä, *data mart*:in käsitteeseen kuuluu, että tiedon lähde on edelleen löydettävissä ja jäljitettävissä takaisinpäin tietovarastoon (Inmon, 1998). Tiedon metatiedot kulkevat tiedon mukana tiedonjalostusketjussa.

Data mart:in mahdollistamat ratkaisut luovat luonnollisen syyn *data mart*:in tarpeelle. *Data mart*:in toiminnassa on kuitenkin myös omat huomionsa. Koska *data mart*:eja on usein monia kuten käyttäjäkuntiakin, on kustannustehokasta, jos kaikki *data mart*:it toimivat hyvin yhteen tietovaraston kanssa. Usein tämä toimii valmiiksi hyvin johtuen *data mart*:in luonteesta tietovaraston jatkeena. Jos *data mart*:in kytkee tietovaraston ohi muihin lähteisiin, saavutaan ongelmaan, jossa *data mart*:ien pitää keskustella monien erilaisten käyttöjärjestelmien kanssa, joka lisää työmäärää. Lisäksi voidaan luoda lisää redundanttisuutta tiedon jalostusputkeen. Nämä ongelmat eivät estä ulkopuolisten tietolähteiden tai tietovaraston ohi tuotavien lähteiden käyttöä, mutta ne on hyvä tiedostaa mahdollisina riskeinä (Inmon, 1998).

2.8 Datan virtualisointi

Kuten aikaisemmin käsiteltiin dataa käsitteenä ja tiedon eri asteita, on myös virtualisoinnille selvennettävä sen merkitystä eri yhteyksissä. Yksi teoreettinen käsite virtualisoinnin yhteydessä on informatisaatio, jossa toiminto viedään digitaaliseen maailmaan, mutta sillä on edelleen yhteys fyysisen maailman toimintaan. Esimerkkinä tästä on virtuaalisilla järjestelmillä ohjattavat systeemit. Teoreettisesti puhdas virtualisointi voi tarkoittaa asian täysin irrottamista fyysisestä maailmasta (Thuemmler, 2017). Tällöin virtualisoinnin esittämä asia on mallinnettu puhtaasti digitaalisessa maailmassa. Taulukkoon 2. on tätä havainnollistettu Health 4.0 -kirjan sivun 42 taulukkoa mukaillen ja lueteltu eri toimintojen kehitystä perustoiminnosta kohti informatisaatiota (Thuemmler, 2017). Datan virtualisoinnin käsitteen osalta tämä merkitysero teoreettisen ideaalin ja käytännön sovellusten välillä on huomioitava. Seuraavaksi havainnollistetaan, miksi täysin virtuaalinen ratkaisu eroaa virtualisoinnin käytännön ratkaisuista.

Jos dataa ja datan jalostusputkea pyrkii ajamaan kohti täysin virtuaalista, teoreettista ideaalia, nykyhetken käytännön toteutukset eivät vastaa käsitteen määrittelyä kaikin osin, sillä osia tiedonjalostusputken kokonaisuudesta on jätettävä virtualisoinnin ulkopuolelle

käytännön syistä. Tämä merkitsee sitä, että esimerkiksi datan sisältävä tietokanta on edelleen tallennettuna levylle fyysisenä kopiona, eikä ainoastaan tallennettuna muistinvaraisesti, eli virtuaalisesti. Käytännön tiedonjalostusputkessa, jonka malli esiteltiin Kuvassa 3, on yleensä käytössä monia eri ohjelmistoja ja systeemejä, joilla voi olla fyysisiä välivarastoja datalle. Osa sovelluksista voi pyöriä vanhemmilla aikaisempien sovellussukupolvien järjestelmillä, eli *legacy*-järjestelmillä. Puhtaasti teoreettista mallia noudattavalla kokonaisvaltaisella datan virtualisointiratkaisulla koko tietoputki olisi mallinnettu virtuaalisesti, jolloin data ja sen käsittelevät operaatiot suoritettaisiin virtuaalisessa ympäristössä alusta loppuun. Olemassa olevien *legacy*-järjestelmien virtualisointi ei välttämättä ole realistista, sillä ne eivät välttämättä tue uudempien sukupolvien tietomalleja ja tiedonsiirtotapoja. Samalla tiedon määrän kasvaessa vaatimukset suorituskyvylle kasvavat. Siirryttäessä uusiin ratkaisuihin, kokonaisuuden tulee pysyä toimivana. Näiden seikkojen vuoksi kokonaisvaltainen virtualisointi on tällä hetkellä lähinnä teoreettinen malli. Osia tiedonjalostusputkesta voidaan kuitenkin virtualisoida. Käytännön sovellukset keskittyvät jonkin tietyn toiminnon virtualisointiin.

Taulukossa 2 on mukailtu Health 4.0 -kirjan taulukkoa, jossa kuvaillaan esimerkinomaisesti eri toimintojen muuntumista kohti virtuaalista. Nämä kehityskulut voivat toimia esimerkkeinä siitä, että virtualisoitu järjestelmä voi olla luontainen kehityskulku toiminnalle. Taulukon kaksi alinta riviä on lisätty kuvaamaan tiedon käytön kahta tärkeää olemusta, tiedon tallentamista ja tiedon hakemista, jotka kuvaavat tiedon jalostamista. Molemmissa voidaan tietyssä mielessä kuvitella datan virtualisointi informatisoiduksi lopputulokseksi, jos kovalevyllisen tietokannan jälkeen esimerkiksi kuvitellaan tulevan pilvipalvelut ja yhtenäiset rajapinnat näihin palveluihin. Samalla tiedon hakeminen virtualisoidusta järjestelmästä toimii samoilla periaatteilla, monen järjestelmän integroidusta kokonaisuudesta rajapinnan avulla, joka osaa yhdistää ne toimivaksi kokonaisuudeksi.

Datan virtualisoinnissa on käytännön ratkaisuissa kyse välikerroksen tuomisesta loppukäyttäjän ohjelman ja tietokannan väliin, jolloin käyttöohjelman ei tarvitse tietää datan sijaintia tai rakennetta (van der Lans, 2012). Van der Lansin mukaan moni muukin tekniikka ja käsite liittyy läheisesti datan virtualisoinnin aihepiiriin ja monet näistä tekniikoista sisältävät samoja periaatteita kuin datan virtualisoinnin yleiset ratkaisut ja tavoitteet. Näitä muita tekniikoita van der Lans mainitsee olevan mm. datan kapselointi, tiedon piilottaminen, datan federaatio ja datan integraatio. Monet näistä suorittavat keskenään samankaltaisia tehtäviä, mutta datan virtualisoinnissa otetaan huomioon näiden toimintojen mahdollisuuksia kokonaisvaltaisemmin ja monet näistä toteuttavat datan virtualisoinnin periaatteen. Täten datan virtualisoinnin ollessa kattotermi, ohjelmistotarjoajat käyttävät erilaisia käsitteitä kuten integraatio tai federaatio virtualisointitermin alla.

Taulukko 2. *Toiminnallisuuden teoreettinen kehityskaari Health 4.0 -kirjan taulukkoa mukaillen* (Thuemmler, 2017)

Toiminto/työkalu	**Mekanisaatio**	**Automatisaatio**	**Tietokoneistaminen**	**Informatisaatio**
Santoor, kielisoitin	Piano	Pianola/automaattipiano	Sähköpiano	Virtuaalinen Piano
Kirjoittaminen	Kirjoituskone	Sähköinen kirjoituskone	Elektroninen kirjoituskone	Tekstinkäsittelyohjelma
Tiskaaminen	Käsikäyttöinen tiskikone	Moderni tiskikone	Mikrosiruohjattu tiskikone	IoT/internet of things, verkkoon yhdistetty ja ohjattu kodinkone
Teollinen valmistaminen	Vaihdettavat osat ja modulaarisuus	Liukuhihnallinen kokoonpanolinja	Robotisoitu valmistus	Teollisuus 4.0/Industry 4.0
Kansio, dokumentti, tieto	Rolodex, indeksikortit	Nauhatallennettu tietokanta	Kovalevyllinen tietokanta	Datan virtualisointi
Tiedon hakeminen selaamalla	Tiedon hakeminen indeksoidusta kokoelmasta, kirjastohyllyt	Tiedon hakeminen kelaamalla tallennettua tiedostoa	Tiedon hakeminen tiedostokokonaisuuksista, tietokannoista, hakulauseet	

Datan federaatiota voidaan pitää datan virtualisointina ja samoin datan integraatiota voidaan pitää datan virtualisointina, mutta tämän perusteella datan federaatio ei tarkoita yhtä kuin datan integraatio, vaan niissä on vivahde-eroja, joista kerrotaan lisää alaluvussa 2.9.1. Rick van der Lansin mukaan datan federaatio tarvitsee aina datan integraation mutta datan integraatiolle datan federaatio on vain mahdollinen datan integraatioväline (van der Lans, 2010). Tärkeimpänä ominaisuutena pysyy kuitenkin mahdollisuus tarjota yhtenäinen rajapinta välikerroksen alla olevaan dataan ja sen käsittelyyn.

Datan virtualisoinnin kehityskulku on läheisesti kytköksissä BI (*business intelligence*) -tekniikoiden kehitykseen, sillä kyseisellä alalla on pyritty ratkaisemaan samoja ongelmia kuin mihin datan virtualisointi tarjoaa ratkaisuja. Esimerkkeinä tällaisista ongelmista ovat datan pirstaleisuus ja monilähteisyys. Datan virtualisoinnin erilaiset ratkaisut voivat siten myös olla vahvasti BI-vaikutteisia, jos ne ovat pyrkineet ratkaisemaan kyseisen alan tyypillisiä ongelmia. Tulevaisuuden mahdollisuudet datan virtualisoinnin ratkaisujen hyödyntämisessä eivät kuitenkaan lepää vain BI-alan harteilla. Terveystiedon lisääntyminen mahdollistaa jopa vallankumouksellisen kehityksen terveydenhuollon palveluissa (Thuemmler, 2017). *Business intelligence* -alalle van der Lans tarjoaa yleisimmäksi datan virtualisointiratkaisuksi virtualisaatiopalvelinta, joka on rakennettu huomioon ottaen BI-alan suuret datamäärät ja SQL-painotteisuus, jotka toimivat hyvin palvelinratkaisussa. Tilanteen mukaan voidaan käyttää muitakin tapoja virtualisoinnin edellyttämän välikerroksen rakentamiseen. Näitä esitellään virtualisointitekniikkoina seuraavassa luvussa.

Koska virtualisointitekniikoita on paljon, on helppo nimetä tuote datan virtualisointiratkaisuksi, jos saavutetaan jokin virtualisoinnin tavoitteista, kuten datan abstraktio, sijainnin piilottaminen tai datan kapselointi monesta eri lähteestä. Tässä saavutaan kuitenkin ongelmalliseen määrittelykysymykseen, jossa pitää pohtia, mitä voidaan pitää datan virtualisointina. Onko tietovarasto datan virtualisointia, sillä sen avulla nähdään kaikki data yhtenäisestä kootusta lähteestä? Onko ohjelmistotuoteperhe datan virtualisointia, sillä kaikki voi näennäisesti tapahtua yhden ohjelmointirajapinnan sisällä, vaikka yksittäiset ohjelmistot eivät virtualisoinnin vaatimuksiin kykenisikään? Näissä tapauksissa täytyy tehdä rajausta termin määrittelyssä.

2.9 Datan virtualisointitekniikat

Datan virtualisointi koostuu välikerroksen muodostamisesta datan ja loppukäyttäjän välille ja siihen kuuluu monia osatekijöitä. Seuraavaksi käsitellään esimerkkeinä erilaisia virtualisointiratkaisuja van der Lans:in mukaan jaoteltuna alaotsikoihin. Eri tekniikoilla saavutetaan eri tavoitteita.

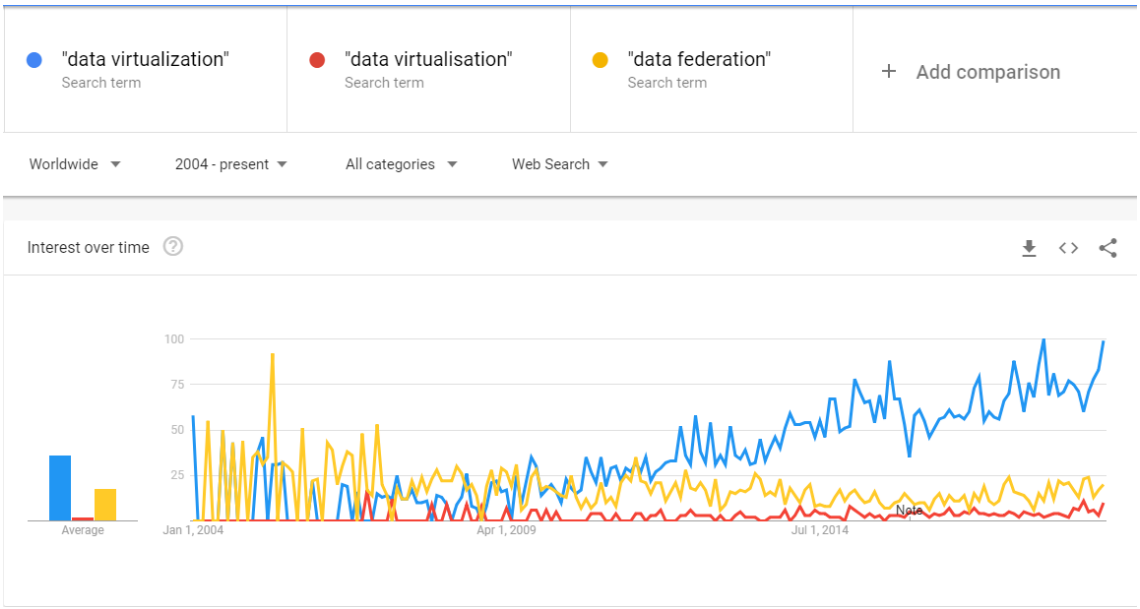
2.9.1 Datan federaatio ja datan integraatio

Datan federaatiosta ja datan integraatiosta mainittiin jo aiemmin, että ne eivät tarkoita keskenään samaa asiaa. Kuitenkin monet palvelut voisivat käyttää kuvauksena toiminnastaan kumpaa tahansa nimeä, ja käytännössä kyseessä onkin osittain vain semanttinen vivahde-ero. Datan federaatiossa tuodaan monesta erillisestä itsenäisestä lähteestä data yhteen järjestelmään. Kun data tuodaan yhtenäiseen järjestelmään, se täytyy muuttaa sopivaan muotoon, jotta erilaiset datat sopivat järjestelmän sääntöihin. Tämän transformaation seurauksena data siis integroidaan kohdejärjestelmään. Tämä kuvaus on lähes sama kuin datan integraatiossa, joka siis sekin tuo datan lähteestä kohdejärjestelmään, mutta van der Lans:in mukaan integraatio ei ota kantaa siihen, onko lähdekantoja yksi vai useita.

Siksi voidaan sanoa, että datan federaatio sisältää aina myös datan integraation, koska muutoin dataa ei saada yhtenäisesti samaan järjestelmään, mutta datan integraatio voi sisältää datan federaation tai olla sisältämättä (van der Lans, 2010).

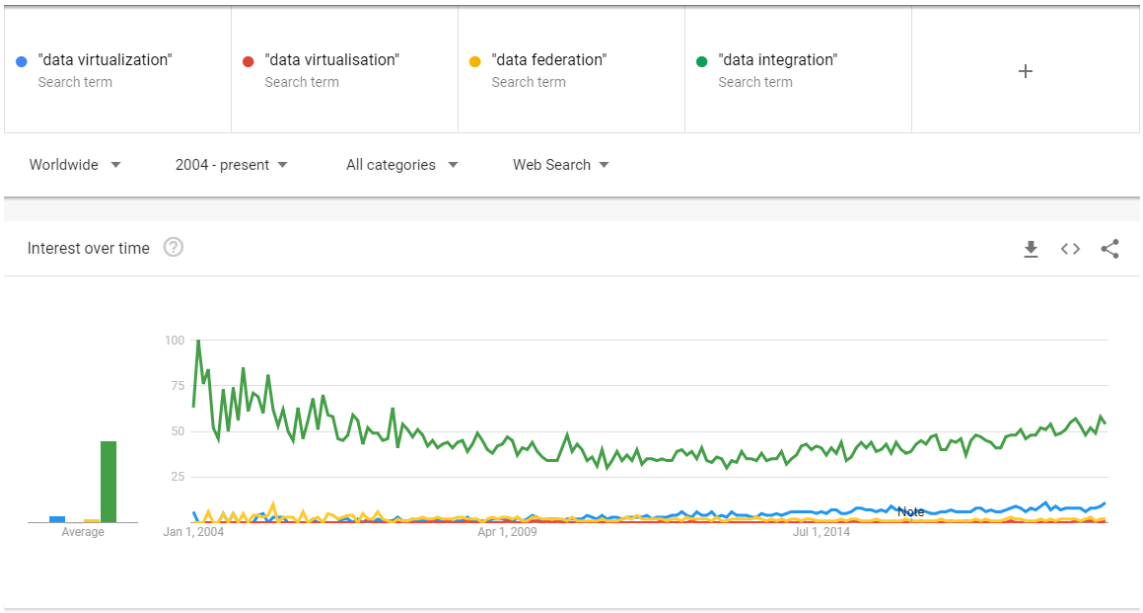
Datan virtualisoinnin kannalta federaatio ja integraatio ovat nimenomaan käytännön ratkaisuissa oleellisia käsitteitä, sillä molemmat suorittavat yhden datan virtualisoinnin tärkeimmistä periaatteista. Data tuodaan yhtenäisen rajapinnan alle. Siksi käytännön ratkaisuissa ja ohjelmistotuotteiden kuvailuissa federaatio tai integraatio on yleisin datan virtualisoinnin käsite ja ohjelmistoratkaisujen tarjoajat käyttävät usein synonyymeina datan virtualisointia, datan federaatiota ja datan integraatiota. Ne eivät kuitenkaan ole synonyymeja kattavasti, sillä datan virtualisointi on käsitteenä moniulotteisempi. Nämä termit ja määritykset, joita tässä työssä on käytetty, ovat myöskin vain yksi näkökulma asiasta, joka perustuu lähinnä alan asiantuntijana pidetyn Rick van der Lans:in kirjoituksiin aiheesta. Termien osalta työn tekemisessä tehtiin rajausta, ja aluksi datan virtualisointia käsiteltiin erillisenä terminä federaatiosta tai integraatiosta, joka vaikuttaa esimerkiksi ohjelmistotarjoajien valintaan, kun vertaillaan datan virtualisoinnin tuottajia. Ratkaisuita vertailtaessa havaitaan kuitenkin, että tuotteissa on eroja, ja federaatio ja integraatio vastaavat usein juuri tiettyyn ongelmaan datanjalostusputkessa. Tämä ongelma on datan hajanaisuus lähteissä. Tämä ei kuitenkaan kata koko datan virtualisoinnin skaalaa, kuten myöhemmin kerrotaan SuperNova-käsitteen yhteydessä, joka merkitsee datan virtualisointipalvelimen käyttöä tietovarastosta ulospäin hajautettaessa dataa käyttötarvetta kohti kuten *data mart*:eissa. Kyseisessä tapauksessa data on jo yhdessä yhtenäisessä järjestelmässä ja sitä hajautetaan kohti loppukäyttäjiä, ja tämäkin on toki datan virtualisointia, mikä toimii esimerkkinä datan virtualisoinnin termin moninaisuudesta.

Yksi tämän diplomityön ongelmista on juuri termien sekalainen käyttö kuvailtaessa ratkaisumalleja. Datan virtualisointiin liittyvää materiaalia ei ole kovinkaan paljon, joka voi osaltaan liittyä tähän seikkaan, että tekniikkaa ja teoriaa, joka on olemukseltaan datan virtualisointia, kuvaillaan toisilla termeillä, joista yleisiä ovat juuri datan federaatio ja datan integraatio. Tämä johtuu siitä, että nämä nimetyt ratkaisumallit sijoittuvat siihen kohtaan tietoputkea, joissa perinteisesti on suuri tarve datan virtualisoinnin kaltaisille ratkaisuille. Tästä voisi kuvitella seuraavan mahdollisesti, että datan virtualisoinnin käsite ponnahtaa pinnalle ja hiipuu sitten, kun tarkemmat sitä hyödyntävät termit ja ratkaisut valtaavat alaa. Yksi mahdollisuus tarkastella tämän toteutumista on Googlen hakutrendien kehitys.



Kuva 4. *Googlen trendi-työkalun kuvankaappaus hakutermien suosiosta* (*Google Trends*, 2019)

Kuvassa 4 on kuvankaappaus Googlen trendi-työkalusta, jolla on haettu hakutermien suosiota ajan kuluessa. Lähtökohdaksi on valittu vuosi 2004. Hakutermeihin on lisätty datan virtualisoinnin kaksi eri kirjoitusmuotoa, sillä ne vaikuttavat hakutermien suosioon. Muoto ”data virtualisation” on käytössä Isossa-Britanniassa ja ”data virtualization” Yhdysvalloissa. Näitä kahta trendiä vasten on lisätty mukaan datan federaatio, joka näkyy kuvassa keltaisella. Kuvaajien kulusta Kuvassa 4 näyttäisi siltä, että datan federaatio on korkeammalla aluksi, mutta datan virtualisoinnin hakutermisuosio kasvaa vuoden 2012 paikkeilla ja jatkaa kasvamistaan sittemmin, eritoten kun huomioidaan, että siniseen kuvaajaan pitäisi lisätä myös punaisen termin suosio. Tämän myötä on todettava, että vaikka datan virtualisointi ei terminä ole tarpeeksi yleistetty ja ajankohtaista tieteellistä materiaalia on niukasti, sen suosio hakuterminä kuitenkin näyttää kasvujohteiselta. Tässä vertailussa ei ole mukana termiä datan integraatio, joka on lisätty mukaan Kuvassa 5.



Kuva 5. *Googlen trendi-työkalun kuvakaappaus hakutermien suosiosta, datan integraatio suhteessa datan federaatioon ja datan virtualisointiin* (*Google Trends*, 2019)

Alemmat kuvaajat Kuvassa 5 ovat samat kuin Kuvassa 4, eli datan virtualisoinnin ja datan federaation kuvaajat. Kuvaajien koosta voidaan havaita, että datan integraatio on paljon suositumpi hakuterminä kuin datan virtualisointi tai federaatio. Tästä voidaan ehkä päätellä, että datan integraatio on tunnetuin termi sellaiselle datan virtualisointiratkaisulle, joka tuo monesta eri lähteestä tiedot yhteen järjestelmään. Datan integraation trendissä on myös näkyvissä samankaltaista varovaista nousua kuin datan virtualisoinnin trendissä. Tämän syy voi piillä mahdollisesti datan käytön viitekehyksessä, eli datan määrän kasvussa, big data:ssa ja datan tuottajien ja hyödyntäjien verkostojen kehittymisessä. Nämä kehityssuunnat osoittavat lisääntyvää kysyntää datan virtualisoinnin kaltaisille ratkaisuille.

2.9.2 Datan virtualisointipalvelin ja virtuaalitaulut

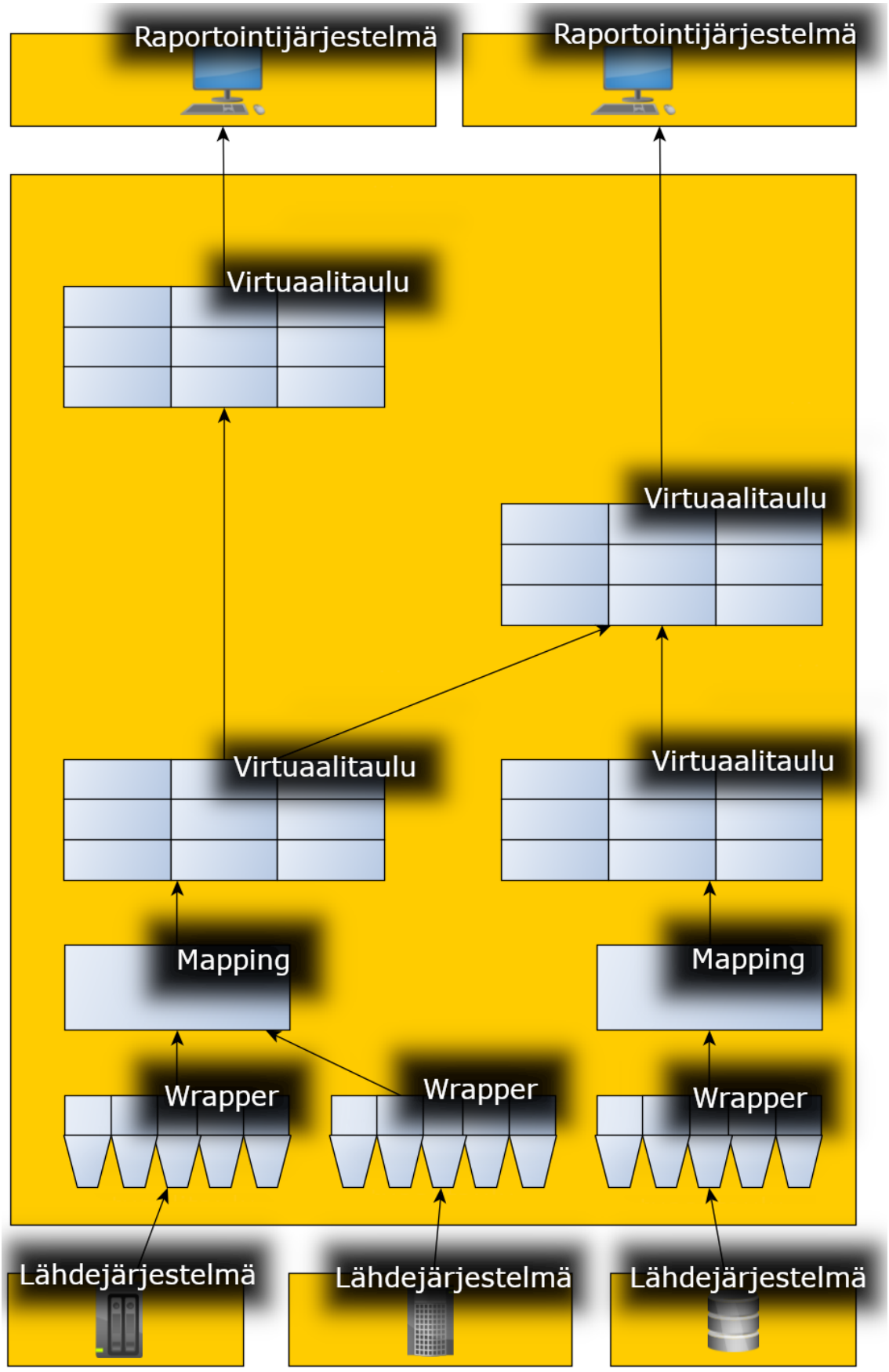
Datan virtualisointipalvelin tarjoaa käyttäjälle näkymän tietokantojen datasta hyväksikäyttäen erilaisia mahdollisia rajapintoja (*Application Programming Interface*, API). Eri API:t tarjoavat peruskäyttökokemuksen, johon loppukäyttäjä on jo tottunut. Eri käyttäjille voidaan myös tarjota erilainen API, riippuen tarpeesta. Virtualisointipalvelin huolehtii datan tarjoamisesta käyttäjille. Ennen datan syöttämistä palvelimelle lähdekanta pitää tuoda mahdollisten transformaatioiden kautta wrappereille. Datan virtualisointipalvelin saa vastaanotettua lähdetietokannoista dataa wrapper-taulujen avulla. Näitä ”käärijä”-taulukoita voidaan kutsua monella muullakin nimellä, mutta kuvaavasti ne ”käärivät” tietokannan lähdetaulusta meta-tiedot, joiden avulla virtualisointipalvelin voi löytää datan, päästä käsiksi siihen ja käsitellä sitä oikealla tavalla ymmärtäen sarakkeiden sisällön ja

tallennusmuodon. Nämä käärijätaulut ovat aina lähdekantakohtaisia, mutta yhdellä lähdeaineistolla voi olla useita erilaisia wrappereita. Wrapperien päälle määritetään virtuaalisia taulukoita, joihin voidaan koostaa tarpeenmukaiset rivit ja sarakkeet tai haluttu tieto wrapper-taulusta. Virtuaalisten taulujen päälle voidaan koostaa toisia virtuaalitauluja. Virtuaalitaulun määritystä pohjalla olevasta taulusta, kuten wrapper-taulusta, kutsutaan kartoitukseksi (*mapping*). Päällekkäisillä virtuaalitauluilla on mahdollista periyttää alempien taulujen ominaisuuksia ylöspäin, jolloin monelle taululle yhteisen datan määrityksen muuttaminen hoituu yhdellä muutoksella näiden yhteisessä juuri-virtuaalitaulussa (van der Lans, 2012). Datan virtualisointipalvelin hoitaa datan federaation ja abstraktion ylöspäin putkessa. Virtuaalitaulut hoitavat datan integraation ja transformaation. Virtuaalipalvelimen muodostamaa kokonaisuutta havainnollistetaan Kuvassa 6, joka myös selkeyttää wrapperien ja *mapping:*in sijoittumista suhteessa virtuaalitauluihin.

Kuvassa 6 eri kerrokset on kuvattu keltaisina laatikkoina. Alhaalla on kolme erillistä lähdetietokantajärjestelmää, jotka muodostavat oman lähdekerroksensa. Ylhäällä on kaksi raportointijärjestelmää, jotka muodostavat siten myöskin oman kerroksensa. Keskellä iso keltainen laatikko kuvaa virtualisointikerroksen toimintaa, kun tietoa pyritään saamaan pohjalta lähteistä kohti yläosaa ja raportteja. Lähdejärjestelmästä data tuodaan virtualisointikerrokseen wrapperin kautta. Wrapperilta tieto jatkaa *mapping*:iin, jota voidaan jo pitää omana virtuaalitaulunaan. Virtuaalitauluja voi luonnollisesti rakennella toistensa päälle, jotta saadaan tarvittavat tiedot ja halutut transformaatiot ja aggregaatiot tehtyä raportointikerrosta varten.

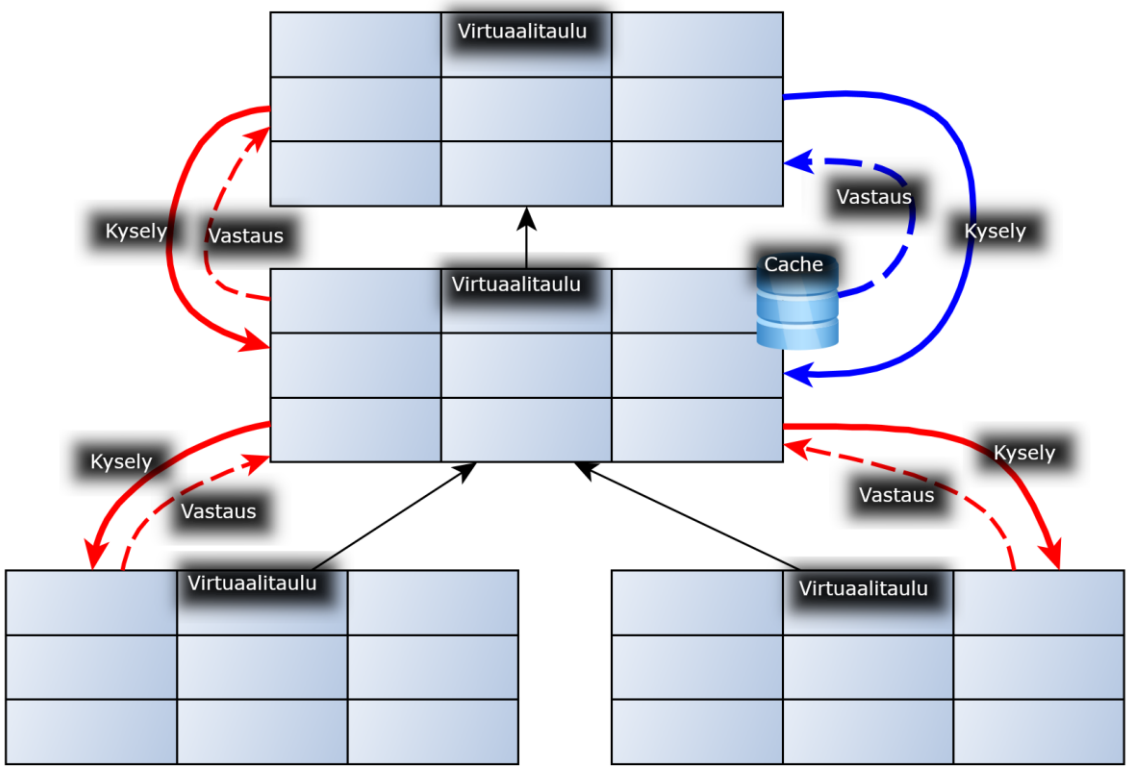
2.9.3 Virtuaalitaulujen tallennus välimuistiin

Tietoa haettaessa täytyy tietenkin suorittaa haku. Tämä haku täytyy määritellä ja syöttää virtualisointipalvelimelle, joka käsittelee haun ja osaa hakea haun parametrien mukaisesti tarvitun datan niiltä tietokannoilta, jotka sisältävät kyseisen datan. Yleensä haut liittyvät päivittäiseen toimintaan, jolloin niitä joutuu todennäköisesti suorittamaan toistuvasti kuten kerran päivässä tai useammin. Tällöin herää kysymys, kuinka kauan tässä prosessissa kestää aikaa. Jotta aikaa ei kulu hakukyselyn käsittelemiseen tietokantojen puolella, voidaan tarvittavia hakuja tallettaa välimuistiin tai levylle riippuen virtualisointipalvelimen ratkaisuista ja tilanteesta. Haku voidaan tämän jälkeen suorittaa nopeasti suoraan tallennettuun välimuistiin ja välttää täten mahdollisesti hidas yhteys ja käsittely pohjalla oleviin tietokantoihin. Avainasemassa välimuistin käytössä on valinta, milloin välimuistiin tallennettu data on päivitettävä pohjalla olevista tietokannoista. Tähän on useita protokollia, jotka sopivat eri tilanteisiin ja hakuihin. Esimerkkejä näistä ovat ajastetut, manuaaliset ja ehdollisesti laukeavat päivitykset (van der Lans, 2012).



Kuva 6. *Virtuaalipalvelinkerros ja virtuaalitaulut van der Lans:in kirjaa mukaillen* (van der Lans, 2012)

Virtuaalitaulujen tallennuksilla tieto on saatavilla nopeasti sopivaan tarkoitukseen muotoiltuna, mikä on käytännössä *data mart*:ien tehtävä. Tiedolle voidaan asettaa rajoituksia, kuinka kauan välimuistillisesti talletettu tieto on validia. Välimuisti voidaan myöskin tallentaa esimerkiksi koostaen, jolloin tieto talletetaan välimuistiin vain, kun sitä kysytään. Tämän jälkeen, jos samaa tietoa kysytään uudelleen ja se vastaa aikalaadultaan tarpeeksi tuoretta dataa, voidaan tieto tarjota välimuistista. Kuva 7 havainnollistaa välimuistin asettumista virtuaalitauluissa.



Kuva 7. *Välimuistiin tallennettu data, ”cache”, ja kysely virtuaalitauluissa van der Lans:ia mukaillen* (van der Lans, 2012)

Kuvassa 7 mustat nuolet kuvaavat virtuaalitaulujen rakentumista toistensa päälle. Punainen kysely lähtee ylhäältä ja kysely jatkuu pohjimmaisiin virtuaalitauluihin asti. Nämä pohjimmaiset virtuaalitaulut hakevat datan lähdekannoista, joita ei kuvaan ole merkitty. Sininen kysely oikealla ylhäällä kuvaa tilannetta, jossa sopiva tieto on välimuistillisesti talletettu ja kyselyyn voidaan vastata tyydyttävästi välimuistin perusteella, jolloin kysely ei jatku alempiin kerroksiin tällä kertaa.

2.9.4 Enterprise Service Bus

Enterprise Service Bus, ESB, on toimintamalli, jossa rakennetaan väylä, jonka kautta kaikki toiminnot kommunikoivat keskenään yrityksessä. ESB hoitaa tiedonkulun oikeaan paikkaan oikeassa muodossa, ja osaa toimia eri ohjelmien välillä tulkkina. Tämänkaltainen pelkkä tulkitsija toimii datan federaation työkaluna, joka toimittaa seuraaville ohjelmille datan integraatiota varten.

2.9.5 Pilvipalvelut

Kun tietokannat talletetaan pilvipalveluihin, käyttäjä voi ottaa yhteyden pilvipalveluun ja toimia sen käyttöliittymän puitteissa. Tällöin käyttäjä ei ota suoraan yhteyttä lähdetietokantaan, eikä käytä lähdetietokannan rajapintaohjelmistoja. Pilvipalvelu toimii siis tässä tapauksessa virtuaalisena rajapintana kooten lähdekantojen datan yhtenäiseen rajapintaan, jota loppukäyttäjä voi käyttää (van der Lans, 2012). Tämänkaltainen ratkaisu on datan virtualisoinnin määrittelyn mukainen. Pilvipalveluita on kuitenkin monia ja kaikki eivät välttämättä tue monipuolisesti erilaisia tietokantaratkaisuja. Käyttötarkoituksen mukaan osan pilvipalveluista voidaan ajatella tekevän datan federaatiota ja integraatiota ilman transformaatiota, jos pilvipalvelu on vain tallennusalusta, joka ei muokkaa dataa toista palvelua varten. Osassa pilvipalveluita käyttötarkoitus voi olla pelkästään suoritustehon hyödyntäminen ilman tallennuskapasiteetin tarvetta.

3. DATAN VIRTUALISOINNIN RATKAISUT

Työn lähtökohdissa rajattiin aiheen käsittely datan virtualisointiin. Teoriaa selvitettäessä ja ohjelmistotarjoajien tuotteita selatessa ilmeni työn alussa vaikeuksia erottaa mikä on datan virtualisointia, sillä termin käyttö oli kirjavaa. Osa palveluntuottajista nimeää tuotteensa datan virtualisoinniksi suoraan ja toiset eivät, mutta tämän perusteella ei vielä voida perustella kumpi ratkaisuista on datan virtualisointia. Kritiikkiä luokitteluista voidaan esittää sen suhteen, että datan virtualisointi voidaan käsittää teoreettisena ideaalina, jota ei käytännössä saavuta yksikään ratkaisu tai laajana ratkaisujen skaalana, joista osia on varmastikin lähes kaikissa ohjelmistoratkaisuissa. Tästä syystä ohjelmistotarjoajia tutkittaessa rajattiin alussa huomio markkinatutkimuksiin, jotka käyttävät termiä datan virtualisointi, vaikka alalla on muitakin termejä, jotka toimivat usein samoissa ongelmissa, kuten luvussa 2 mainitut datan federaatio ja datan integraatio. Näistä termeistä on erikseen olemassa markkinatutkimuksia ja usein mukana voikin olla samoja toimijoita, kuin on mainittu datan virtualisoinnista tehdyissä markkinatutkimuksissa.

3.1 Ohjelmistotarjoajat

Yhtenä diplomityön menetelmänä on selvittää tämänhetkisten palveluntarjoajien tasoa ja tuotteiden vertailua. Toimiala on suuressa kasvu- ja muutosvaiheessa, jolloin palveluntarjoajien kirjo vaihtelee. Suuret toimijat nousevat mainoksillaan ja läsnäolollaan esille jo kirjallisuudessa, mutta markkinatutkimukset tunnistavat muitakin markkinoilla nousevia toimittajia. Markkinantutkimuksia on monia ja monien yritysten tekeminä. Tässä työssä on käytetty kahta markkinatutkimusta, jotka käyttävät käsitettä datan virtualisointi.

Forresterin raportti listaa 13 merkittävintä toimijaa alalla (Yuhanna, Leganza ja Perdoni, 2017). Nämä palveluntarjoajat ovat DataVirtuality, Denodo Technologies, IBM, Informatica, Looker, Microsoft, Oracle, Pitney Bowes, Red Hat, Rocket Software, SAP, SAS ja TIBCO Software. Raportissa on mainittu paljon samoja toimijoita, joita mainittiin aiemmassa vuoden 2015 raportissa, jolloin toimijoiksi mainittiin seuraavat 9 palveluntarjoajaa: Cisco Systems, Denodo Technologies, IBM, Informatica, Microsoft, Oracle, Red Hat, SAP ja SAS Institute (Yuhanna, Owens ja Cullen, 2015). Uusia toimijoita vuoden 2017 listauksessa ovat DataVirtuality, Looker, Pitney Bowes ja Rocket Software sekä Cisco Systemsin datavirtualisaatiotoiminnan ostanut TIBCO Systems.

Gartner listaa omassa markkinaopasteessaan palveluntoimittajina olevan Actifio, Cisco, Data Virtuality, Denodo, Gluent, IBM, Informatica, Information Builders, OpenLink Software, Oracle, Primary Data, Progress, Red Hat, Rocket Software, SAP, SAS ja Stone Bond Technologies (Zaidi, Beyer ja Jain, 2017). Seuraavaksi käsitellään mainittuja palveluntarjoajia ja heidän datan virtualisointiratkaisujaan.

3.1.1 Actifio

Actifio on vuonna 2009 perustettu yhdysvaltalainen IT-firma, jonka datan virtualisointiratkaisun pohjana heidän tuotteessaan on virtuaalinen dataputki, VDP. VDP toimii alustana, jonka päälle Actifion muut ratkaisut rakentuvat ja jota hallitaan Actifion API:n kautta. VDP:n perusajatuksia ovat muun muassa datan kaappaaminen alkuperäisessä muodossaan, self-service yksinkertaisena, datan tarjoamisen mahdollistaminen kopiovirtualisoinnilla, alustakeskeisyys ja nopea saatavuus dataan. Actifion datan virtualisaation ratkaisu on kopiodatan virtualisointi. Tämä merkitsee, että datasta säilytetään fyysisenä ”kultainen kopio”, josta voidaan edelleen kopioida lisää kopioita virtuaalisesti, jotka päivittyvät tämän alkuperäisen kopion mukaan. Kopioita tarvitaan kuorman ja eri käyttötarkoitusten vuoksi, mutta fyysinen kopiointi vaatii resursseja eri tavoin kuin virtuaalinen ratkaisu.

3.1.2 Cisco

Cisco on perinteinen toimija tietotekniikan alalla, perustettu vuonna 1984 Yhdysvalloissa. Ciscon datavirtualisoinnin tuotteet ja palvelut myytiin vuonna 2017 TIBCO Systemsille, ja tarkempi kuvaus tarjonnasta kuvaillaan TIBCO Systemsin alla.

3.1.3 Data Virtuality

Data Virtuality on vuonna 2012 perustettu tietopalveluratkaisuja tarjoava yritys. Data Virtuality:n datan virtualisoinnin ratkaisut lähtevät liikkeelle virtuaalisesta tietoputkesta lähdekantoihin. Tämän virtuaalisen kerroksen hakema data voidaan tallentaa haluttuun tietovarastoon. Tuotteena on joko pelkkä putkiratkaisu tai kattavampi tietovaraston mallintamisessa avustava paketti, joka myös käsittelee dataa analyysityökaluille päin sopivaksi.

3.1.4 Denodo

Denodo Technologies on vuonna 1999 perustettu yhdysvaltalainen yritys. Denodon ratkaisu datan virtualisointiin muodostuu Denodon alustamallisesta Denodo Platform -tuotteesta, joka toimii virtualisointikerroksena datan lähteiden ja datan kuluttajien välillä. Denodon tuote tavoittelee hakuoptimointia ja datan siirtelemisen tarpeettomuutta.

3.1.5 Gluent

Gluent on vuonna 2014 perustettu yhdysvaltalainen yritys. Gluentin datavirtualisointituotteen tarkoitus on ehkäistä datan siiloutumista ja tarjota data yrityksen sisäiseen väylään ilman ETL-putkien lisäämistä.

3.1.6 IBM

IBM on vuonna 1911 perustettu yhdysvaltalainen IT-alan yritys, jolla on kokonsa puolesta suuri määrä tuotteita, työntekijöitä ja asiakkaita. IBM:n datan virtualisointiin liittyvät palvelut ovat lukuisia ja suuri osa niistä liittyy datan integraatioon. Puhdasta virtualisointia edustavat tuotteet taas toimivat esimerkiksi keskustietokoneissa. IBM:n tuoteperhe on niin laaja, että datan virtualisoinnin määrityksen mukainen välikerros on mahdollista rakentaa tuotteiden yhdistelmänä, vaikka mikään yksittäinen tuotenimi ei välttämättä tekisi kaikkia latausputken toimintoja puhtaasti datan virtualisoinnin periaatteilla.

3.1.7 Informatica

Informatica on vuonna 1993 perustettu yhdysvaltalainen yritys. Informatica on sijoittunut vahvasti markkinatutkimusraporteissa ja sen markkinaosuus on suuri. Informatican tuotetarjonta on myös laaja ja valintaa vaikeuttavat useiden tuotteiden samankaltaiset käyttömahdollisuudet.

3.1.8 Information Builders

Information Builders on vuonna 1975 perustettu yhdysvaltalainen yritys. Information Builders:in datan virtualisointiin liittyvät ratkaisut liittyvät heidän tarjoamaansa alustaan, jossa Omni-Gen niminen ympäristö suorittaa datan hallintaa *master data management* tyylillä, jossa taustalla on *master data record*. Tämä merkitsee sitä, että *master data record* toimii ikään kuin ”kultaisena kopiona” datasta. Sen data on siivottu duplikaateista ja se toimii vertailupisteenä muille datakopioille. Tavoitteena on muun muassa datan integraatio.

3.1.9 Looker

Looker on vuonna 2011 perustettu yhdysvaltalainen yritys, jonka datan virtualisointituotteena toimii Lookerin oma alusta, joka perustuu datan ”upotukseen” (*embedding*), jonka avulla datan kuluttajille tarjotaan näkymä ajankohtaiseen dataan ja samalla painopiste on vahvasti ajan tasalla olevan datan visualisoinnissa.

3.1.10 Microsoft

Microsoft on vuonna 1975 perustettu yhdysvaltalainen yritys. Microsoftin datan virtualisoinnin ratkaisu on rakennettu Microsoftin olemassa olevien tuotteiden, tässä tapauksessa Azuren, sisälle, yhdistämällä niiden toimintoja, jotta saavutetaan datan virtualisoinnin tavoitteita. Tavoitteena on kyselyjen kuormituksen jakaminen, jolloin kyselyn käsitteleminen hajautetaan ja alkulähde ei rasitu vaan pilvipalvelu transformoi ja integroi kyselyn tuloksen.

3.1.11 OpenLink Software

OpenLink Software on vuonna 1992 Isossa-Britanniassa perustettu yritys, jonka tuotteet toimivat standardien mukaisina välipalveluina yhdistäen asiakkaiden tuotteet. OpenLinkin datan virtualisointiratkaisu on OpenLink Virtuoso -alusta, jonka se mainostaa olevan koko yrityksen laajuinen ”kytkentärasia”, jonka välityksellä eri ratkaisut voidaan yhdistää eli integroida.

3.1.12 Oracle

Oracle on vuonna 1977 perustettu yhdysvaltalainen yritys, joka on perinteinen toimija tietokanta-alalla. Oraclen tuotelinja on valtava, josta datan virtualisointiin eniten liittyy Oraclen datan integraatio-työkalu.

3.1.13 Pitney Bowes

Pitney Bowes on Yhdysvalloissa vuonna 1920 perustettu alun perin postilogistiikkaan keskittynyt yritys. Pitney Bowesin datan virtualisointiratkaisu liittyy heidän tarjoamaansa Spectrum-alustaan, jossa datan federaatio ja integraatiotoiminnot auttavat yhdistämään useita tietolähteitä.

3.1.14 Progress Software

Progress Software on vuonna 1981 perustettu yhdysvaltainen yritys, jonka datan virtualisointiratkaisut perustuvat heidän tarjoamaansa DataDirect-palveluun ja sen konnektoreihin, jotka ovat yhteyksien luomista varten eri tietokantojen ja pilvipalvelujen välillä. Kyseessä on välipalvelu, joka toimii muiden tarjoamien palvelujen datan välittäjänä. Progress Software:n pilvidatan federaatiopalvelu on yhdistetty *Hybrid Data Pipeline*:ksi, joka yhdistää pilvipalvelujen toiminnallisuuden federaatiotyökalujen kanssa siten, että tämänkaltainen hybridi-ratkaisu osaa toimia palomuurin kanssa.

3.1.15 Red Hat

Red Hat on vuonna 1983 perustettu kansainvälinen yritys, joka tuottaa avoimen lähdekoodin ohjelmistoja. Red Hatin datan virtualisoinnin ratkaisu on JBoss-välipalvelualustalla toimiva datan integraatioon perustuva kerroksellinen ohjelmisto ja työkalurypäs, joka sisältää datalähteiden konnektorit, datan transformaation ja datan syöttämisen kohdealustoille. Se käyttää hyväkseen datan virtualisoinnin välimuistillisia näkymiä ja datan abstraktiota yhtenäisen rajapinnan taakse.

3.1.16 Rocket Software

Rocket Software on vuonna 1990 perustettu yritys, jonka pääkonttori on Yhdysvalloissa. Rocket Softwaren tuotteet on suunniteltu keskustietokoneita varten. Eritoten Rocket Softwarella on kumppanuussuhde IBM:n kanssa ja Rocket Software tarjoaa ratkaisuja esimerkiksi IBM:n keskustietokoneita varten. Rocket Softwaren datan virtualisointiratkaisut perustuvat IBM:n keskustietokoneella pyörivään z/OS-käyttöjärjestelmään, jossa voidaan ajaa datan virtualisointipalvelinta, joka integroi keskustietokoneen sisältämän datan ulkopuolisten tietolähteiden kanssa ja saattaa ne yhtenäisen rajapinnan alle, jota loppukäyttäjät osaavat käyttää.

3.1.17 SAP

SAP on vuonna 1972 Saksassa perustettu yritys, joka keskittyy tuotannonohjausjärjestelmien tekemiseen. SAP:n datan virtualisointiratkaisu koostuu SAP HANA data-alustasta, joka suorittaa muun muassa datan federaation ja integraation yhdessä SAP:n työkalujen kanssa.

3.1.18 SAS

SAS Institute on Yhdysvalloissa vuonna 1976 perustettu analyysiohjelmistojen valmistaja. SAS:n datan virtualisointiratkaisut liittyvät heidän tuotteissaan datan federaatioon SAS:n omalla federaatio-ohjelmistolla, joka toimii yhdessä SAS:n muilla alustaohjelmistopaketeilla.

3.1.19 Stone Bond Technologies

Stone Bond Technologies on vuonna 2001 perustettu yhdysvaltalainen yritys, jonka datan virtualisointiratkaisu perustuu Enterprise Enabler -nimiseen alustaohjelmistoon, joka sisältää mm. datan transformaation ja tietolähteiden yhdistämisen. Transformaatiotyökalulla Stone Bond mainostaa muun muassa takaisin lähteeseen päin kirjoittamista. Virtualisoinnin seurauksena Stone Bond mainitsee myös käsitteen virtuaalinen tietovarasto (*Logical Data Warehouse*), jossa tieto saa säilyä alkulähteillään. Looginen tietovarasto on käsitteenä kuitenkin hämärä, sillä Rick van der Lansilla on oma määrittelynsä termille ja hän mainitsee myös käytännössä kaikkien nykyaikaisten tietovarastojen olevan Logical Data Warehouse -ratkaisuja, verrattuna perinteiseen tietovarastoon.

3.1.20 TIBCO Software

TIBCO Software on Yhdysvalloissa vuonna 1997 perustettu väliohjelmistojen tarjoaja. TIBCO Software on myös tehnyt useita hankintoja ostaessaan muita ohjelmistotuottajia, kuten CISCO:n datan virtualisoinnin osaston. Tämä näkyy TIBCO:n datan virtualisoinnin

ratkaisussa, joka on datan integraatiota ohjelmistopaletilla, joka on vahvasti modulaarinen. Hankintojen lisääminen ja osaamisen hyödyntäminen TIBCO:n tuotteissa on selvemmin toteutettavissa, kun modulaariseen työkalusettiin voidaan täydentää yksittäisen osatekijän sisältöä. CISCO:n tuotteisiin kuuluneen CISCO Information Server:in yhteydessä tehdyssä *white paper* -dokumentissa esiteltiin Rick van der Lansin johdolla SuperNova -malli *data mart*:ien virtualisointiin, kun käytetään *data vault 2.0* -konseptia. Sittemmin SuperNova-konseptia ei mainita TIBCO:n esitteissä, mutta *data mart*:ien virtualisointi on osana TIBCON datan virtualisoinnin ratkaisua.

3.2 Ohjelmistotarjoajien vertailu

Kun ohjelmistojen tarjoajia ja tuotteita vertaillaan, on otettava huomioon monia asioita. Palveluntarjoajien käsitteet voivat vaihdella, kun puhutaan nimenomaan datan virtualisoinnista. Vaikka käsite olisi sama, voidaan se ymmärtää eri tavoin markkinointipuheissa. Kohderyhmien ja käyttötarkoitusten erilaisuus vaikuttaa myös tuotteisiin. Tuotteiden kypsyys *life cycle* -ajattelussa vaikuttaa muun muassa siten, että osaan tuotteista datan virtualisointi saattaa olla myöhemmin syklissä lisätty ominaisuus tai tavoite, kun taas toisissa tuotteissa datan virtualisointi on voinut olla käsitteenä jo alusta lähtien ja tuote on rakennettu tätä käsitettä silmällä pitäen. Näistä ja muista seikoista johtuen kaikki vertailut eivät ole suoraan keskenään toisiinsa vertailukelpoisia vaan niissä täytyy pitää mielessä palvelun taustat. Jotta vertailun laajuus kattaisi niitä seikkoja, joita kaikista tarjoajista pystytään selvittämään ja pysyisi siten vertailukelpoisena, on valittu vertailtavaksi melko suppea joukko ominaisuuksia. Valittuja ominaisuuksia ovat tässä vertailussa yrityksen perustamisvuosi ja yrityksen tuotepaletin tarkastelu. Tuotteista on selvitetty millä määritelmällä yritys kuvailee tuotteitansa ja mihin tarkoitukseen. Yrityksen tuotetta on myös tarkasteltu sen pohjalta, perustuuko tuote alustapohjaiseen ohjelmistoon, joka toimii omassa ekosysteemissään ja onko tuote tehty nimenomaan yrityksen omaa alustaa varten. Osa tuotteista on puhtaasti väliohjelmistoksi kehitettyjä, osa on kehitetty toimimaan nimenomaan omassa alustassaan.

Taulukko 3. *Datan virtualisointiratkaisujen tarjoajien koostetaulukko*

Tarjoajat	**Perustettu**	**Tuotepaletti, määritelmä, putki**	**Alustapohjainen**	**Väliohjelmisto vai oma alusta**
Actifio	2009	Kopiovirtualisointi, *self-service*, virtuaalinen Dataputki	Kyllä	Väli
Data Virtuality	2012	tietoputki, lähteeltä tietovarastoon ja eteenpäin	Ei	Väli
Denodo	1999	Virtualisointikerros lähteeltä eteenpäin	Kyllä	Väli
Gluent	2014	Siilojen purku, integraatio sisäiseen väylään	Ei	Väli
IBM	1911	Monituote, integraatio keskustietokoneissa	Kyllä	Oma
Informatica	1993	Monituote	Kyllä	Väli
Information Builders	1975	Omni-Gen, *master data management, data integration*	Kyllä	Väli
Looker	2011	Lookerin oma alusta, datan visualisointi datan kuluttajille	Kyllä	Väli
Microsoft	1975	Azure-pohjainen, kyselyhajautus	Kyllä	Oma
OpenLink Software	1992	Yrityslaajuinen kytkentä, integraattori	Kyllä	Väli
Oracle	1977	Monituote, datan integraatio	Kyllä	Oma
Pitney Bowes	1920	Tietolähteiden yhdistäminen	Kyllä	Väli
Progress	1981	Konnektorivetoinen, yhteydet, pilvidata	Ei	Väli
Red Hat	1983	Monituote, OpenSource, konnektorit, näkymät, abstraktio	Kyllä	Väli

Rocket Software	1990	Keskustietokoneiden integraatio ulkopuolisten kanssa	Kyllä/Ei	Oma
SAP	1972	Monituote, integraatio ja federaatio	Kyllä	Oma
SAS	1976	Monituote, integraatio ja federaatio	Kyllä	Oma
Stone Bond Technologies	2001	Transformaatio ja yhdistäminen	Kyllä	Väli
TIBCO Software	1997	Monituote, modulaarinen, SuperNova, data martien virtualisointi	Kyllä	Väli
Muut				
Cisco	1984	TIBCO osti Ciscon datan virtualisoinnin		
SuperNova	2014	*Data mart* -virtualisointikonsepti, Cisco, siitä TIBCO:hon		
Primary Data	2013	Sulkeutui 2018		

Taulukon 3 loppuun on lisätty datan virtualisointiratkaisut, jotka ovat lopettaneet tai vaihtaneet omistajaa, sekä SuperNova, joka oikeastaan on suunnittelukonsepti, joka myöskin on vaihtanut omistajaa. Edelleen toimivia tarjoajia taulukossa on 19, joista viisi on perustettu 2000-luvun jälkeen ja näistä kolme on perustettu 2010-luvun jälkeen. Suhteellisesti voidaan siis sanoa, että suurin osa toimijoista on pitkän linjan palveluntuottajia ja datan virtualisointiratkaisujen kehittäminen on vain jatkoa näiden toimijoiden muuhun tuotepalettiin. Datan virtualisointi tekniikkana on, kuten todettua teoriaosuudessa, lähinnä tiettyjen integraatiotekniikoiden ja menetelmien yhdistymistä saman termin alle. Näiden tekniikoiden pohja ei välttämättä ole mitenkään uusi, jolloin osalla toimijoista on vanhastaan ollut datan virtualisointia suorittava tuote tai datan virtualisoinnin ratkaisua vaativa tuote. Tätä taustaa vasten, datan virtualisointi terminä on kasvattanut suosiota vuoden 2012 jälkeen, kuten nähtiin hakutrendien osiossa teoriapuolella. Tämä trendikasvu johtuu osittain datan virtualisointia käsittelevän kirjan julkaisusta vuonna 2012 (van der Lans, 2012), joka on toiminut tekniikan merkkipaaluna. Tämä tausta luo kontrastin datan virtualisoinnin pohjaratkaisujen vanhemman alkuperän ja toisaalta termin uutuuden välille. Taulukosta voidaan havaita, että uusia toimijoita on verrattain vähän ja yksi uusi toimija on lopettanutkin. Koska yritykset on valittu markkinatutkimusten perusteella kiinnostavista toimijoista, tämä ei välttämättä ole yllättävä tulos, sillä uusilla tekijöillä ei ole vanhan ja

asemansa vakiinnuttaneen toimijan vahvuutta markkinoilla, jolloin uudet ja pienet toimijat eivät välttämättä näytä kiinnostavilta markkinatutkimusten näkökulmasta, varsinkin, jos niiden tuote ei ole kypsä. Uusilla toimijoilla ei myöskään ole luultavasti muuta tuotepalettia täydentämään tarjontaa. Niinpä suurin osa palveluntarjoajista on vanhoja toimijoita, joilla on paljon muitakin tuotteita ja palveluja ja datan virtualisointi on joko lisätty näiden tuotteiden päälle tai osa vanhoista tekniikoista yhdentyy kohti datan virtualisoinnin käsitettä.

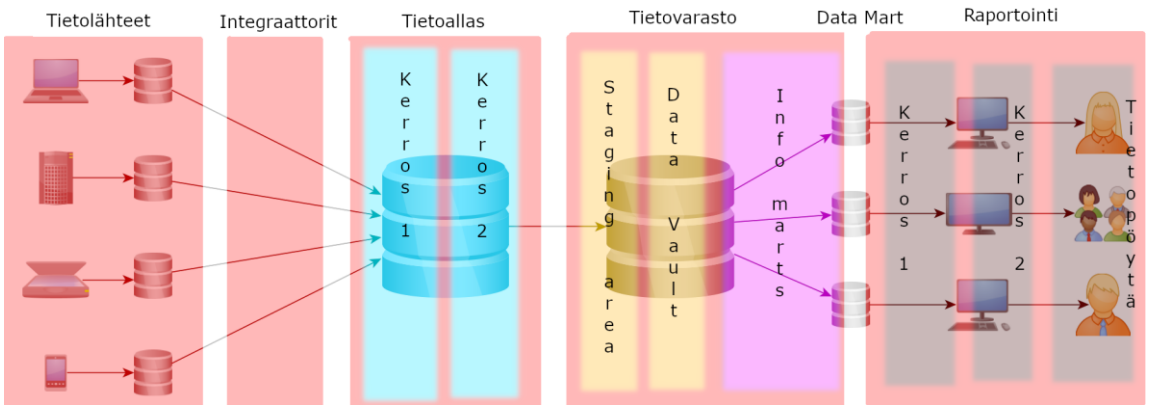
Palveluntarjoajat ovat siis suureksi osaksi vanhoja toimijoita. Siksi ei ole yllättävää, että suurin osa ratkaisuista perustuu jollekin alustalle, joka usein liittyy muihin tuotteisiin ja ratkaisuihin, joita yritys tarjoaa. Alustatalous on myös laajennettavuudessaan kasvava nykyhetken trendi. Alustoissa on kuitenkin eroja ja osa toimijoista pyrkii tuottamaan alustan, joka toimisi väliohjelmistona eri rajapintojen kanssa. Toiset alustat puolestaan pyrkivät lähtökohdiltaan vakiinnuttamaan tarjoajan omaa alustaa, jonka juuret ovat jossakin muussa toiminnassa, kuten keskustietokoneissa tai tuotannonohjausjärjestelmässä ja tähän alustaan lisätään omat integraatiotyökalut ja datan virtualisoinnin ratkaisu yleisesti. Taulukon luokitukset ovat kuitenkin osittain tulkinnanvaraisia, sillä ne perustuvat yritysten omaan materiaaliin ja julkisesti saatavilla olevaan tietoon yritysten tuotteista ja toiminnasta.

4. DATAN VIRTUALISOINTI 2M-IT:N TIETOPUTKEN YHTEYDESSÄ

Tämä luku käsittelee nykyistä tiedonjalostusputkea 2M-IT:n näkökulmasta. Nykyhetken tiedonkäsittely perustuu taustalla oleviin tarpeisiin ja tarjolla oleviin ratkaisumalleihin ja tuotteisiin, joilla voidaan vastata tarpeisiin. Lähteenä nykyhetken kuvaukselle toimivat asiantuntijahaastattelut, joissa on haastateltu kahta 2M-IT:n BI-asiantuntijaa tietovaraston ja tietoaltaan toiminnasta kesällä 2018. Kysymysrungot haastatteluista on lisätty liitteenä. Lisäksi lisänäkökulmaa tarjoavat vapaamuotoisemmat keskustelut 2M-IT:n henkilöstön kanssa ja WhereScape:n konsulttien kanssa.

4.1 2M-IT:n tietoputken kuvaus

Tietoputken toiminnassa pohjalla on valtava määrä hyvin erityyppistä dataa. Operatiivisessa käytössä data voidaan säilyttää alkulähteessään, mutta strategisessa käytössä dataa joudutaan siirtämään ja muokkaamaan, jotta se sopii raportointityökaluille ja dataa saadaan koostettua monesta lähteestä kattavasti. Lähteet voivat olla järjestelmien sisäpuolisia tai ulkopuolisia lähteitä. Data on luonteeltaan suureksi osaksi terveyspalveluihin liittyvää dataa, jonka avulla voidaan esimerkiksi hallinnoida terveydenhuoltopalveluita. Terveystiedon erikoisluonne näkyy mm. tarkassa käyttöoikeus- ja luvitusprosessissa. Työn alkuperäistä dataputken kuvaa (Kuva 3) mukaillen Kuvaan 8 on sovitettu 2M-IT:n datan jalostusputken vaiheita.



Kuva 8. *2M-IT:n datan jalostusprosessin osatekijöitä sovitettuna dataputken perusmalliseen kuvaan*

Kuvassa 8 punaiset laatikot kuvaavat prosessin vaiheita. Osa prosesseista on jaettu vielä osatekijöihin, joita on kuvattu erivärisillä laatikoilla punaisten laatikoiden sisällä. Vasemmalla kuvassa ovat tiedon lähteet ja tietokannat, jotka voivat olla ulkoisia tai sisäisiä. Tietolähteitä on monia ja niiden formaatit ovat erilaisia.

Jotta dataa saadaan hallitusti eteenpäin lähteistä, on käytössä integraattoreita, joiden voidaan ajatella toimivan samalla konnektoreina ja transformaation alkutekijöinä. Integraattorin on luotava yhteys tietolähteeseen ja tulkittava sen dataformaatti eteenpäin. Integraattorit varmistavat yhteyden toimivan lähteeseen ja tekevät alustavaa työtä mm. dataformaattien määrittelyssä ja johdattavat tiedon lopulta Hadoop-pohjaiseen *data lake*-malliin. Integraattorityökaluja on käytössä useita tarpeen mukaan, kuten Scoop, Oozie tai Apache Kafka.

Dataa johdetaan tietoaltaaseen (*data lake*), joka kuvassa on jaettu vielä kahteen osaan sinisillä laatikoilla. Itse tietoaltaan toiminta perustuu Clouderan tuottamaan Hadoop-pohjaiseen ohjelmistoratkaisuun. Ensimmäiseen kerrokseen data tulee integraattorien kautta lähteistä, jolloin integraatiota suoritetaan aktiivisesti sitä mukaa kun dataa saadaan. Tietoaltaassa tieto jäsennellään edelleen metatietojen kera raakatiedosta kohti taulutettua rakennetta. Tässä prosessissa tiedolle varmistetaan yksilöivä tunnus, ID, ja muun muassa avainten, taulujen ja kenttien kuvaukset

Tietoaltaasta dataa siirretään tietovarastoon (*data warehouse*). Kuvassa tietovarasto on jaettu kolmeen osaan. Tietoaltaan ja tietovaraston välissä on suoritettava ETL-lataus, jossa tieto ensin annetaan *staging area*:an, joka muodostaa ikään kuin eteisen, josta tieto tiedonsiirtoprosessien mukaan talletetaan tietovarastoon. Tietovaraston lataamisessa pyritään käyttämään latausautomaatiotyökalua, joka on WhereScape:n tuote. WhereScapen tuoteperheessa WhereScape RED ja WhereScape 3D muodostavat kokonaisuuden, joka yhdessä mahdollistaa latausten mallintamisen ja suorittamisen. Kuvassa nämä WhereScapen tuotteet ovat kaksi ensimmäistä osaa tietovaraston kolmesta osasta. WhereScape 3D:n avulla luodaan latauksen malli ja RED:n avulla voidaan suorittaa edellisen mallin mukainen ETL-lataus. Näiden lataustyökalujen käytöllä data saadaan sopimaan Data-Vault 2.0 -mallin mukaiseksi, mikä helpottaa datan jatkojalostusta eteenpäin tiedonjalostusputkessa. Samalla muokkausmahdollisuudet helpottuvat, jos vain osa mallista täytyy muuttaa. Lataustyökaluilla datan siirtämistä ja muokkaamista voidaan osittain automatisoida, jolloin on tiedossa, miten ja missä muodossa dataa on käsiteltävä. Näitä toimintoja voidaan mallintaa ja seurata automaatiotyökaluilla. Automatisointi helpottaa perustoimintojen suorittamista, jolloin voidaan keskittyä suunnittelemiseen ja mahdollisten ongelmatapausten selvittämiseen. Latauksen parametrit ja operaatiot voidaan tallentaa malliksi. Tätä mallia voidaan hyödyntää, kun suoritetaan samaa latausta uudelleen. Seuraavat lataukset ovat yksinkertaisempia toteuttaa myös muokattuina, kun pohjana on sama malli. Samalla kun toimitaan mallin mukaan, on lopputulos aina sama, mikä tarkoittaa laadun

kannalta tasaista lopputulosta. Latausparametreissa voi olla tulkinnanvaraisuutta ja latausten suorittajilla voi olla eroavia työtapoja, jotka saattavat näkyä latauksissa ilman mallia.

Tietovaraston jälkeen tietoa voidaan jäsennellä näkymille ja tauluille käyttötapausten mukaan *data mart*:eihin, joista raportointityökalut osaavat hakea haluamansa tiedon siinä muodossa, kuin se niiden tarvitsemana on muokattuna *data mart*:issa. Kuvassa tämä toiminallisuus on kuvattu osana tietovarastoa, *information mart*:eina, jotka toimittavat osan *datan mart*:ien töistä. *Information mart*:eissa tietovaraston tietoa aletaan jäsennellä käyttötapausten mukaan ja luodaan tietomalli ja latausasetukset kohti seuraavia tietoputken osia. Näkymiä ja tauluja aletaan erotella ja niitä jaetaan kohti loppukäyttäjiä, joiden tarpeet vaativat erilaisia tietoja.

Kuvan 8 viimeinen osuus, raportointi, on jaettu kolmeen osaan. Niistä kerros 1 jatkaa *information mart*:ien työtä ja siinä tietomalli testataan ja siihen lisätään semanttinen tietomalli. Semanttisessa tietomallissa tietoon yhdistetään sen merkitys tai tarkoitus reaalimaailmassa. Tämä merkitsee, että jatkossa tietoyksiköitä kuten näkymiä ja tauluja voidaan käsitellä niiden luonnollisessa merkityksessä, kuten näkymä jonkin terveyspalvelun tuottajan kaikista kuvantamisen toimista, sen sijaan, että näkymä olisi merkitykseltään esimerkiksi vain taulujen A ja B unioni leikkauksena taulusta C, jotka ovat peräisin operatiivisesta kannasta kohdista yksi ja kaksi. Raportointikerros 2 jatkaa tiedon käsittelyä *business intelligence* -sovelluksille sopivaksi raportointia varten. Tämä kerros sisältää ratkaisun testauksen ja käyttäjien koulutuksen ja lopulta tuotantoon viennin. Viimeinen kerros raportointiosuudessa on tietopöytä, jolloin raportointityökalut toimittavat tiedon loppukäyttäjien päätteisiin.

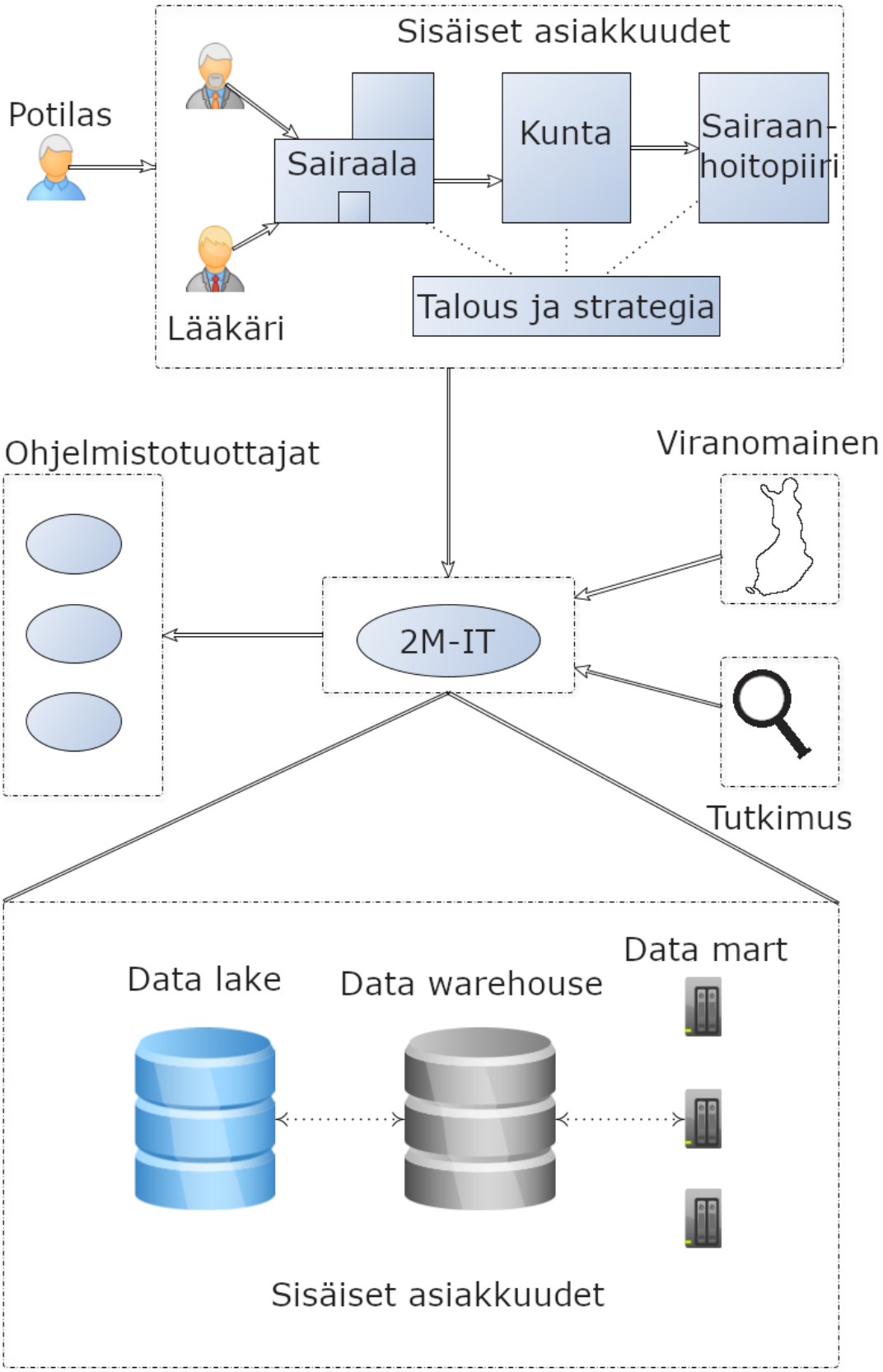
4.2 Asiantuntijahaastattelut

Tässä kappaleessa tiivistetään asiantuntijahaastatteluiden sisältö. 2M-IT:n tietovarasto-asiantuntijaa haastateltiin kasvotusten 13.6.2018 ja tietoallasasiantuntijaa kokousohjelman välityksellä 20.6.2018. Haastatteluihin oli rakennettu runko, joka noudatti samaa mallia molemmissa haastatteluissa. Tämä runko on lisätty liitteeksi diplomityön loppuun. Luonnollisesti runko ei sisällä jatkokysymyksiä tai syventäviä kysymyksiä, jotka ovat aihealueittain erilaisia. Rungon ulkopuolisten kysymysten sisältöä avataan, kun sellainen sattuu tässä käsittelyssä kohdalle. Haastattelun taustana oli pitkälti sama tavoite kuin diplomityön alkuperäisessä tarvesuunnittelussa, eli datan virtualisoinnin hyödynnettävyys tietovarastopohjaisissa tiedonhallintaratkaisuissa. Tätä taustaa vasten haastatteluissa pyrittiin selvittämään kuvausta nykyhetken työnkulusta ja tekniikoista ja miten nämä mahdollisesti peilautuisivat datan virtualisoinnin käsitteen kanssa. Luonnollisesti myös itse tilaajayrityksestä, 2M-IT:stä, oli saatavissa lisätietoa, jota onkin jo hyödynnetty muun muassa 2M-IT:n tietoputken kuvauksessa.

Haastattelun alussa selvitettiin asiantuntijoiden taustaa ammatissa ja käytännön työtä. Asiantuntijoilla on korkeakoulututkinnot ja vankka kokemus alalta. Tietovaraston asiantuntijalla on kokemusta tietokannoista ja tiedonhallinnasta, kun tietoaltaan asiantuntijan kokemus painottuu *business intelligence* ja tiedonhallinnan aloille. Molemmissa haastatteluissa tuli ilmi käytännön työn projektiluontoisuus ja sen edellyttämä asiakasläheinen työskentely. Projektit lähtevät yleensä asiakkaan tarpeesta tai uusien työkalujen käyttöönotosta. Asiakkaan puolella loppukäyttäjiä on erilaisia ja siten tarpeetkin ovat erilaisia. Asiakkuussuhteita onkin hahmoteltu tarkemmin Kuvaan 9. Kuvassa 2M-IT sijoittuu keskelle ja nuolet kuvaavat asiakkuussuhdetta.

Haastatteluissa käsiteltiin myös 2M-IT:n toiminnan sijoittumista terveysalalla. Potilas on terveydenhuollon asiakas. Lääkäri tai muu hoitohenkilöstö on yleensä potilaan kontaktina sairaalaan, terveysasemaan tai muuhun hoitopisteeseen. Terveydenhuollon eri toimijat on kuvassa huomioitu sisäisinä asiakkuuksina. Sairaalat toimivat kunnissa, jotka kuuluvat yleensä johonkin sairaanhoitopiiriin. Näihin liittyvät hallinnolliset elimet, jotka seuraavat taloutta ja strategiaa, kun johdetaan terveydenhuollon toimintaa. Strategisen tason seuranta vaatii jalostettua tietoa, jota 2M-IT:n datan jalostusputki pyrkii tarjoamaan. Lisäksi päivittäisessä toiminnassa tarvittava tieto raportteihin saadaan jalostettua dataputken avulla. 2M-IT toimii yli kymmenen maakunnan ja yli kymmenen sairaanhoitopiirin alueella. Julkisomisteisena osakeyhtiönä, nämä asiakassuhteet ovat 2M-IT:lle yleensä omistaja-asiakkaita. Lisäksi tutkimuskäyttö ja viranomaistoiminta voivat olla asiakkaina. 2M-IT:n sisäistä toimintaa on avattu lähinnä dataputken ydintoimintojen ympäriltä, jotka toimivat yrityksen sisällä ikään kuin sisäisinä asiakkuuksina, jotka asettavat toisilleen tarpeita ja tarjoavat edellytyksiä toistensa toiminnalle. Käytännössä nämä osiot suorittavat siis tiimityöskentelyä keskenään koko putken pituudelta. Koska 2M-IT hyödyntää tarjolla olevia ohjelmistoratkaisuja, on yritys itse luonnollisesti näiden ohjelmistotuottajien asiakas.

Nykyhetken tarpeista ja tavoitteista kysyttäessä selvitettiin tarkemmin asiakkaiden vaatimuksia ja asiakassegmentin sisäistä jakaantumista erilaisiksi loppukäyttäjiksi mm. analyytikkojen, hallinnon ja potilastyön tekijöiden kesken. Ohjelmistotyölle on yleinen haaste, että saadaan määriteltyä asiakkaan kanssa mitä asiakas oikeasti haluaa ja mitkä ovat mahdollisuudet eri toiminnoissa. 2M-IT:n tapauksessa asiakkaalla on yleensä selkeä tavoite saada tietty luku tai tieto näkyviin raportille ja tarve on siten selkeästi määritelty. Toisaalta asiakkaalla ei välttämättä ole tietoa, miten tähän lukuun päädytään. Riippuen näkökulmasta, projekteissa ratkaistavat haasteet ovat halutun tiedon selvittäminen ja tiedon löytäminen sekä yhteyksien toimiminen tietoputken sisälle. Asiantuntija voi esitellä asiakkaalle mitkä kaikki menettelytavat ovat mahdollisia työkaluissa.



Kuva 9. *2M-IT:n asiakassuhteita*

Projektin edetessä on avuksi, kun ollaan yhteydessä asiakkaaseen ja asiakkaan haluama visio tarkentuu projektin edetessä. Näistä seikoista johtuen asiakas on hyvä mieltää osaksi kehitystiimiä. Silloin voidaan esitellä jo varhaisessa vaiheessa jotain toimivaa väliratkaisua tai viimeistelemätöntä *end-to-end* ratkaisua asiakkaalle, jotta ratkaisua voidaan työstää kohti varsinaista tarvetta ja toiminnallisuutta. Vaikka asiakas ei koe tarvetta tietää prosessia tarkemmin tai nähdä viimeistelemättömiä väliratkaisuita, näiden sisällyttäminen asiakassuhteeseen parantaa asiakkaan lopullisen tavoitteen saavuttamista.

Nykyhetken tekniikan työkaluista kysymykset käsittelivät tekniikoiden kirjoa ja toimivuutta edellä mainittuihin tavoitteisiin nähden. 2M-IT:n tekniikoita ja työkaluja onkin jo esitelty 2M-IT:n dataputkesta kertovan kappaleen yhteydessä. Data voi kulkea hieman eri työkalujen kautta riippuen mistä järjestelmästä data on lähtöisin ja mihin järjestelmään se on menossa. Lähdejärjestelmien moninaisuus tuo haasteita *legacy*-järjestelminä, eli järjestelminä, jotka voivat olla jo vanhoja ja jotka eivät aina tue uusia työmenetelmiä. Esimerkiksi data voi olla lähdejärjestelmässä sellaisessa muodossa, että se sisältää keskenään eriäviä merkintätapoja samoista asioista tai datan tietomallia joudutaan muokkaamaan runsaasti, jotta se saadaan tasalaatuisena luetuksi eteenpäin. Työkaluina toimivat mm. lataus-, kanta- ja raportointiohjelmistot. Lähdejärjestelmät ovat kuitenkin isoja ja osia kokonaisuudessa, jonka tulee pysyä toimivana, jolloin lähdejärjestelmiä harvoin voi muokata. Työssä ratkaisun toteuttaminen voi edellyttää taustalla olevien tietojen ja tiedonkeräysjärjestelmien päivittämistä, jos haluttua pohjatietoa ei ole saatavilla. Datalle on toki olemassa myös tiettyjä malleja, jotka olisivat suositeltavia ja joissa data on sopivassa muodossa. Tällöin jatkotyökalut osaavat käsitellä dataa helpommin vähemmillä muokkauksilla. Jotta tieto saadaan jalostettua loppuun asti, on sitä käsiteltävä ja selviteltävä, jotta sille voidaan tehdä oikeat operaatiot. Ja toki vaikka työkalut sopisivat yhteen ja data olisi niiden kanssa yhteensopiva, ohjelmistotyölle yleinen virheiden ja häiriöiden korjaus sekä manuaalinen koodaus kuuluvat työhön.

Lopuksi kysyttiin tulevaisuudennäkymistä. Näkemys on, että dataa tulee useammista eri lähteistä, joissa on erityyppistä dataa yhdisteltäväksi ja analysoitavaksi. Nykyisten tiedonjalostussyklien lisäksi kaivataan lisää näitä yhdisteleviä ratkaisuja, jotka muodostavat isompia kokonaisuuksia. Nykymallissa dataa olisi paljon saatavilla, mutta kokonaiskuvan muodostaminen voi olla vaikeaa. Tämän kokonaismallin lisäksi analyysin tavoitteena voi olla vielä enemmän ennustava suunta. Ei ainoastaan nykyhetken raportointia, vaan myös tulevaa trendien kehitystä. Työkalujen sisällä mallit ja automatisoidut prosessit auttavat luomaan yhdenmukaisia lopputuloksia tiedon prosessoinnissa. Toisaalta vielä ei ole saavutettu sitä potentiaalia, mitä tekoäly ja koneoppiminen voivat tuoda. Kun tekniikka kypsyy, herää myös asiakkaiden ymmärrys mahdollisuuksista, mikä johtaa tarpeiden muuttumiseen ja lisääntymiseen. Asiakkaiden määrä myös kasvaa tulevaisuudessa.

4.3 WhereScape:n konsulttien näkemys

WhereScape:n konsulteilta kysyttiin myös näkemystä datan virtualisoinnista. Näin saatiin hieman alalla toimivien palveluntarjoajien kuvaa aihealueesta. Tietotekniikan alalla yleistä on uuden ratkaisun hakeminen ja siltä odotettavat mahdollisuudet ja parannukset. Ei kuitenkaan ole olemassa ”hopealuotia”, kuten Fred Brooksin artikkeli totesi jo vuonna 1987 (Brooks, 1987). Odotukset mullistavasta ja kertaluokkaa työtä parantavasta tekniikasta ohjelmistopuolella eivät siis realisoidu samalla tavalla kuin tietokoneiden fyysisen suorituskyvyn puolella. Datan virtualisointi noudattaa samaa kaavaa, jossa ohjelmistotarjoajat käyttävät hyväkseen uuden termin taakse kerääntynyttä jännittynyttä odotusta ja mahdollisuuksien kirjoa. Kokonaisvaltaisena ratkaisuna datan virtualisointi ei lunasta paikkaansa, mutta sillä voi olla etua lisäosana ja työkaluna. Datan virtualisointia voidaan käyttää liikutettaessa dataa kohti tietovarastoa tai sitä voidaan käyttää hyödyntämään tietovarastoa tietolähteenä. Datan siirto-operaatioissa datan virtualisointia voidaan käyttää datan muokkauksessa ennen varsinaista siirtoa ja siistimisessä sekä järjestelemisessä. Markkinoilla on olemassa käyttäjiä, jotka hyödyntävät datan virtualisoinnin ratkaisuja. Virtuaalisen kerroksen yksi etu on, kuten mainittu jo monesti, yhtenäinen kerros loppukuluttajaa kohden. Tällöin voidaan tehdä muutoksia taustalla ilman, että se häiritsee huomattavasti käyttäjiä tai muuttaa käyttökokemusta. Tämän ja datan siirron operaatioiden yhteydessä datan virtualisoinnista voi olla hyötyä, kun latauksia on määrällisesti paljon, mutta suurten erien latauksessa datan virtualisointi ei tuo uutta etua.

WhereScapen tuoteperheessä SuperNova-liitin on olemassa. SuperNova on datan virtualisoinnin ratkaisu *data mart*:ien paikalle ja aikaisemmin mainittiin, että SuperNova on oikeastaan suunnittelukonsepti, jolloin palveluntarjoaja voi toteuttaa sen haluamallaan tavalla oman ohjelmistonsa sisällä.

4.4 Denodo demo

Jotta saadaan käsitys, miten datan virtualisointi käytännön sovelluksissa toimii, suoritettiin yksinkertainen kokeilu ohjelmistolla, joka kuvailee itseään datan virtualisointiratkaisuksi. Kokeilulla haluttiin testata, miten tekniikan lupaamat kyvyt helppoudesta ja nopeudesta toteutuvat käytännön työssä. Kokeilulle ei asetettu tavoitteita suorituskyvystä tai erillisen testidatan analysoinnista, vaan hyödynnettiin ohjelmistotarjoajan omia resursseja ja ohjeita.

Käytännön datan virtualisoinnin ratkaisuista eräs helposti lähestyttävä on Denodo. Denodo ei markkinatutkimusten perusteella kuulu markkinajohtajiin vaan se on luokiteltu haastajaksi, mutta kuitenkin melko kypsäksi tuotteeksi, jolla on kyvykkyyksiä. Denodon tuotteista löytyy ilmaisversioita vuoden lisenssillä ja Denodon sivuilla on kattaus ohjeita ja harjoituksia. Nämä seikat johtivat siihen, että vaikka Denodo ei välttämättä ole markkinajohtaja tai kyvykkäin toimija, Denodon tuotetta on matala kynnys kokeilla ja suorittaa demoja ilman, että käyttäjän pitäisi sitoutua ostamaan tai maksamaan lisenssimaksua.

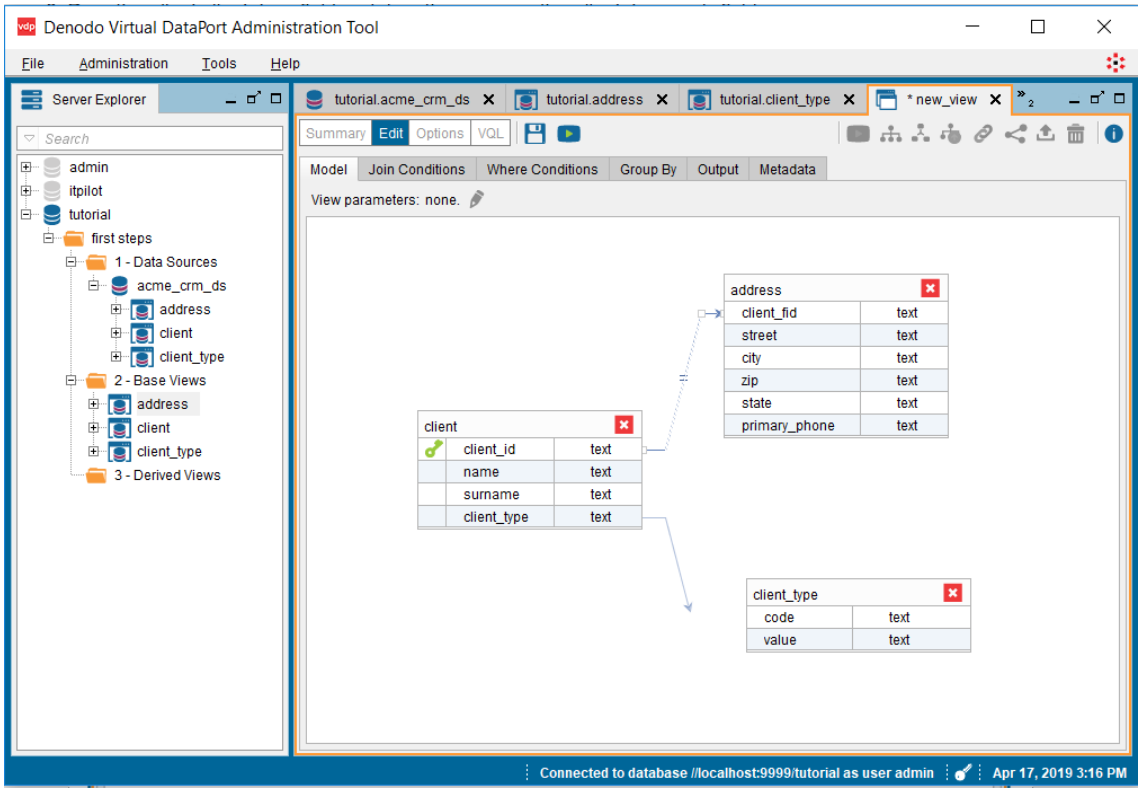
Toki Denodollakin on täysimittainen tuote ilmaislisenssin lisäksi, jolla on lisätty kyvykkyyksiä, mutta ilmaislisenssin avulla suoritettu kokeilu riittää demonstroimaan datan virtualisoinnin ratkaisua siinä määrin kuin ohjeet ja Denodon omat testitietokannat edellyttävät.

Ennen kuin tietokantoja voidaan testata Denodon sovelluksessa, pitää tietenkin luoda tietokanta. Denodon ohjeistuksissa on annettu valmiiksi .sql-tiedosto, joka muodostaa sen mukaisen *schema*:n tauluineen kuin opetusohjeissa käytetään. Tämä tietokanta luodaan MySQL:n avulla, jossa *schema*:n mukaisen tietokannan luonnissa helpottaa MySQL Workbench. Tässä vaiheessa työtä kovin intuitiivisesta prosessista ei ollut kyse, sillä versioissa ja riippuvuuksissa piilee mahdollisia epäsopivuuksia, jolloin esimerkiksi Denodo:n tarjoaman esimerkki–*schema*:n luontiskriptiä tuli hieman muuttaa, jotta se kääntyy MySQL 8.0:n mukaiseksi. Tämä toimii esimerkkinä siitä, että skriptit ja automatisoidut toiminnot kyllä nopeuttavat työnkulkua ja luovat yhteneviä lopputuloksia, kunhan skripti tai toiminto toimii ilman ongelmia. Kun lähdetään ratkomaan skriptin tai toiminnon automaation ongelmia, voidaan päätyä tilanteeseen, jolloin pitää miettiä, onko lopussa odottava automatisoitu työpanos suurempi kuin bugin tai skriptin työstämiseen käytetty aika ja työpanos.

MySQL oli käytössä esimerkissä, sillä Denodo käyttää esimerkkiohjeissa konnektoria, joka JDBC:n (*Java Database Connectivity*) avulla osaa ottaa yhteyden tietokantaan. Näitä konnektoreita on toki paljon muitakin, jotka sopivat Denodon tuella heidän tuotteisiinsa. JDBC-konnektorilla ja ajureilla Denodo voi yhdistää tietokantaan, joka pyörii harjoitusohjeissa samalla koneella. Tämäntyyppisten konnektoreiden toiminta on yksi datan virtualisointituotteiden myyntivalteista. Tässä kohtaa on huomioitava, että kyseiset konnektorit eivät ole uniikkeja datan virtualisointiratkaisuille vaan ne ovat käytännössä standardi, joka pätee koko tietokanta-alalla ja datan virtualisointi perii tämän toiminnallisuuden sillä perusteella, että se on yleisesti levinnyt tapa toimia. Kun konnektori toimii, se toimii ajurina, joka osaa välittää tiedonkulun eri rajapintojen yli. Tämänkaltaisten yhteyksien luominen käsin on paljon aikaa vievä operaatio, joten jos rajapintaan on valmis konnektori, säästää se huomattavan paljon työaikaa. Tämä oletus perustuu siihen, että konnektori toimii ilman lisäsäätöjä. Jos kyseessä on erikoistapaus, johon ei ole valmista konnektoria tai valmiit konnektorit eivät toimi automaattisesti, työn tehokkuuden kasvun mahdollisuus ei realisoidu.

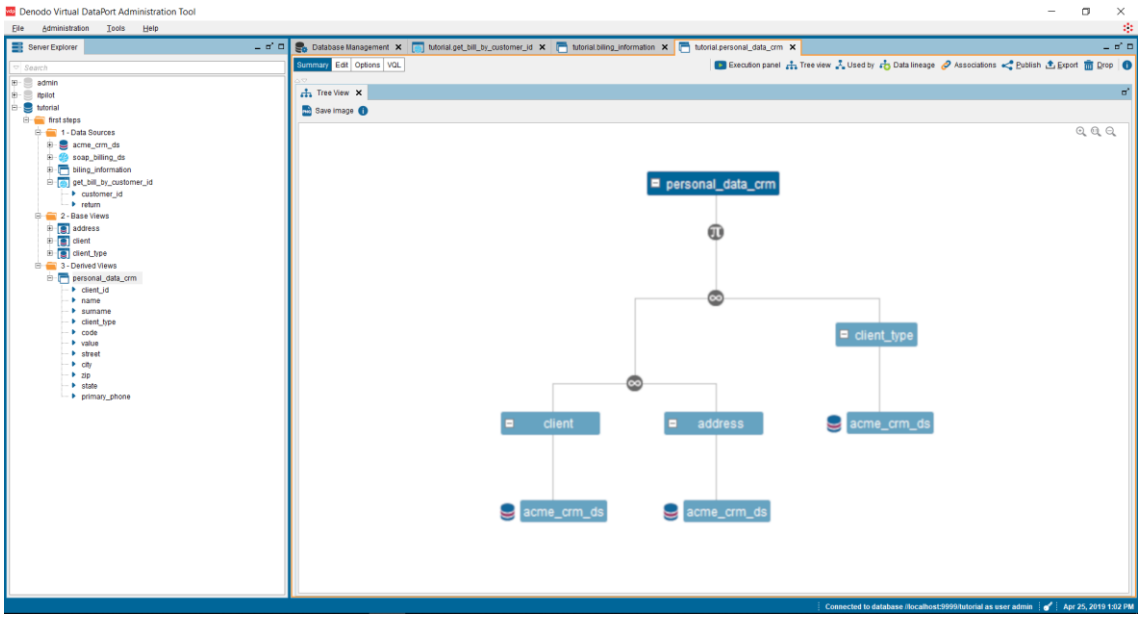
Kun yhteys tietokantaan toimii, Denodon omassa sovelluksessa voidaan suorittaa näkymien muokkauksia ja hakuja tietokannan tiedoista. Kun tietoa kysytään Denodon omassa sovelluksessa, se hakee tiedot lähdekannasta. Lähdekannan tietoja voidaan yhdistellä näkymiksi, jotka toimivat käytännössä samoin, kuin SQL-näkymät, mutta niitä ei talleteta tietokantaan, vaan Denodon omaan sovellukseen. Denodo Express ei silti talleta tietokantojen tietoa mihinkään fyysisesti vaan kyselee ne tarvittaessa ja suorittaa metatietojen ja riippuvuuksien mukaiset toimenpiteet, jotta tieto voidaan saada näkymiin halutussa muodossa ja halutuilla nimillä. Denodon oma kerros on siis vain yhteyksien ja riippuvuuksien

verkosto, jonka kerrokset ikään kuin vetelevät sätkynuken naruista ilman, että ne sisältäisivät itse nukkea. Tietokantojen sisältämät taulut, jopa eri tietokantojen sisältämät taulut, on yksinkertaista linkittää toisiinsa vierasavaimen avulla, jonka suhteen kertominen onnistuu sekin visuaalisesti, josta on esimerkkinä Kuva 10.



Kuva 10. *Denodon taulujen vierasavaimet*

Kuvassa 10 esitetään ohjelman näyttämä muokatun näkymän rakentaminen useista näkymistä. Ylempi viiva on vedetty *client*-taulun *client_id* -kentästä *address*-taulun *client_fid*-kenttään ja viivalle on määritetty suhteeksi yhtäsuuruus, sillä molemmat sisältävät asiakas ID:n. Alempi viiva on *drag-and-drop* -toiminnolla hiirellä vedetty *client_type* -kentästä, mutta sitä ei ole vielä vedetty loppuun asti. Lopulta alempi viiva vedetään hiirellä suhteeksi *client_type*-näkymän *code*-kenttään. Tämä toiminto on tässä kuvailtu näin, jotta havainnollistetaan Denodon ohjelmiston intuitiivista toimintaa ja toimintojen välivaiheellista suorittamista, jossa suhdetta ei tarvitse vetää yhdellä komennolla loppuun asti. Tässä esimerkissä näkymät ovat peräisin samasta tietokannasta, mutta sama periaate pätee eri kantojen yhdistelyssä. Näitä eri kannoista peräisin olevia näkymiä ja yhdistettyjen näkymien yhdistelmiä havainnollistamaan Denodo tarjoaa puunäkymän. Kuvassa 11 on puurakenne, jossa on samoja äsken tehtyjä näkymien suhteita, jotta voidaan hahmottaa missä tauluissa näkymien alkuperät ovat.



Kuva 11. *Denodon näkymien puurakenne*

Fyysisen datan tallentamattomuus on yksi datan virtualisoinnin ominaisuuksista, jotka tekevät siitä ketterän. Versioita ja muokkauksia on nopea muuttaa, kun samalla ei tarvitse käsitellä varsinaista datamassaa. Tähän seikkaan kiteytyy kuitenkin yksi käytännön ongelmista, joka asettaa taustavaatimuksen ongelmattomalle toiminnalle. Virtualisointiratkaisulla on oltava yhteys datalähteeseen, jotta se voi näyttää sen sisältämää dataa. Jos datan lähde on väliaikaisesti pois päältä tai saavuttamattomissa, virtualisointikerrokset eivät pysty kysymään sen sisältämää dataa. Yksi ratkaisu tähän on virtuaalisten taulujen välimuistillinen tallentaminen, joka on esitetty jo teoriaosuudessa. Denodon ohjelmassa välimuistillisille tauluille asetetaan *Time-To-Live*, TTL, joka kertoo kirjaimellisesti, kuinka kauan taulun sisältämä tieto on validia, ennen kuin sen sisältö on ”kuollutta” ja se pitää hakea uusiksi, jos sitä kysytään. Lisäksi tallennustoiminto voi olla osittainen tai täysi. Osittaisessa tallennuksessa, tieto kysytään ensi haulla lähteestä ja seuraavilla hauilla välimuistista, jos välimuisti sisältää kysytyn datan. Täydessä tallennuksessa koko lähde kysytään valmiiksi välimuistiin ja haut kohdistuvat vain välimuistiin.

Denodon ohjelmisto pyrkii olemaan intuitiivinen toiminnoissaan. Yksi esimerkki tästä on *drag-and-drop*-toiminnallisuus, jossa elementtejä voidaan editoida ja yhdistellä tiputtamalla niitä kursorilla oikeisiin kohtiin työtilassa. Tämä helppokäyttöisyys nopeuttaa ohjelmiston oppimista ja selkeyttää käyttäjälle, miten hän voi saavuttaa sen toiminnon, jota hän hakee. Helppokäyttöisyys on myös sisäänrakennettuna niihin konnektoreihin, joita Denodon ohjelmisto osaa hyödyntää. Tästä esimerkkinä hakupyyntö, joka olettaa haettavan lauseen lisäksi muuta syötettä, jotta se menee läpi. Denodo:n ohjelma ymmärsi kyseisen funktion vaativan tietyntyyppiset syötteet hakuehtoihin ja täydensi ne automaattisesti muista näkymistä, jotka olivat liitettyjä Denodoon.

5. JOHTOPÄÄTÖKSET JA TULEVAISUUDENNÄKYMÄT

Yhteenvetokappale on jaettu kahteen osioon, joista ensimmäisessä pohditaan datan virtualisoinnin ratkaisun käyttöönottamista nykyisessä toimintatilanteessa 2M-IT:n ympäristössä. Soveltuvuutta on pohdittu koko tiedonjalostusputken osalta. Lopulta on todettava, että datan virtualisoinnin käsitteen mukaisia menettelytapoja on jo jokapäiväisessä toiminnassa mukana, sillä kattotermin alle mahtuu monia käsitteitä. Tästä syystä on vaikea sulkea jotakin ratkaisua täysin ulkopuolelle, että juuri tämä toiminto ei ole millään tavalla datan virtualisointia. Lopuksi on kuvailtu tulevaisuudennäkymiä terveyspalveluiden toimintaympäristössä.

5.1 Soveltuvuus tiedonjalostusputkeen

Kuten aikaisemmin on mainittu, datan virtualisoinnin ratkaisuiden ei ole tarkoitus yleensä korvata koko datan jalostusputkea datan lähteistä saakka, koska se ei ole käytännönmukaista, joten voidaan pohtia tiedonjalostusputken datan prosessointiosia ja datan virtualisoinnin soveltumista niihin. 2M-IT:n tiedonjalostusputkessa *data vault 2.0* -mallin mukainen tietovarasto on isossa roolissa esimerkiksi historiatiedon säilömisessä, joka on oleellinen osa strategista seurantaa ja pitkäaikaista suunnittelua tiedon käyttäjien puolella. Datan virtualisointiratkaisuiden tarkoitus ei yleensä ole korvata tietovarastoa, joten datan jalostusputken prosessointiosuuden kokonaisvaltainen vaihtaminen datan virtualisointiin ei sovellu korvaajaksi siinäkään mielessä, sillä osia tiedonjalostusputkesta on jätettävä virtualisoinnin ulkopuolelle.

Datan lähteiden moninaisuudessa datan virtualisointia on hyödynnetty monessa palvelussa. Tiedonlähteistä saatavan datan määrä on 2M-IT:n tapauksessa suuri ja WhereScape:n konsulttien mukaan datan virtualisoinnin käyttö tämänkaltaisessa tapauksessa, jossa eräajot ovat suuria, ei välttämättä hyödy kaikista datan virtualisoinnin mahdollisuuksista. Vastakkaisena vertailukohtana voidaan pitää pieniä eräajoja, joita tehdään monia ja usein. Suurten eräajojen haastetta voidaan kuvailla kaistanleveyden riittämättömyydellä. Kun suuri määrä dataa pitää ajaa eräajosta läpi, se vie aikaa, ja eräajoille on asetettu aikatavoitteita. Viimeistään se asettaa eräajon pituudelle kestorajoituksen, kun erän pitää toki olla ajettuna ennen seuraavan ajon aloitusajankohtaa. Käytännössä tämä aikaraja on paljon tiukempi, sillä datan pitää olla käytettävissä jo ajojen välisenä aikana. Ohjelmistoille tämäntyyppisen kaistan ajaminen kapasiteettia vasten asettaa vaatimuksen palvelun skaalautuvuudesta, sillä tiedonsiirron määrällinen tarve ei ole vakio. Jos ohjelmistoa voidaan suorittaa hajautetulla konekannalla, skaalautuvuus yleensä tulee mahdolliseksi. Aina

tämä ei kuitenkaan ole mahdollista ja ohjelmistot voivat vaatia lisenssiehdoissaan maksuja suorittavien koneiden mukaan. Hajautettu suorituskyky käytännössä nykyaikana tarkoittaa yleensä pilvipalvelujen käyttöä. Pilvi itsessään voi olla yhteisön sisäinen tai ulkoisen palveluntarjoajan.

Datan virtualisoinnin yksi visio ja mahdollisuus olisi työn tehokkuuden nousu, jos tekniikka lunastaa kaikki lupauksensa. Yksi houkuttelevimpia visioita olisi, jos virtualisointiohjelmisto osaisi automaattisesti koota lähdekannoista metatiedot ja tuoda datan jonkinlaisella rakenteella järjestelmään sisään. Samalla tämä lähdetietokannan ja siihen yhdistyvän virtualisointiohjelmiston välinen yhteys toimisi automaattisesti, eli virtualisointiratkaisu ymmärtäisi lähdekantaa ja sen järjestelmää automaattisesti. Käytännössä tämä ei toteudu ja näiden yhteyksien rakentaminen ja riippuvuuksien ja tiedon varmistaminen säilyy edelleen työvaiheena yleisimmissä datan virtualisointiohjelmistoissa. Automatisoitu ratkaisukaan ei kykene automaattisesti toimimaan, jos kyseessä on erikoistapaus tai harvinaisempi tietokanta ja datamuoto. Terveysdatassa kyseiset erikoistapaukset eivät ole mitenkään harvinaisia. Kyvykkäämmät sovellukset sisältävät vahvoja moottoreita, joihin on voitu ohjelmoida myös näitä erikoisempia ratkaisuja. Näissä sovelluksissa hinta seuraa kyvykkyyden kasvua vahvasti, jolloin ne eivät vielä ole kustannustehokkaita vaihtoehtoja varsinkin, jos ei ole takuuta siitä, että ne kykenisivät automaattisesti suorittamaan työvaiheita ilman manuaalista säätämistä ja siten korvaamaan muita tuotteita tai työvaiheita.

Kokonaisuudessaan voidaan todeta, että datan virtualisointi on yksi yhdentymissuunta nykyajan tekniikoille. Sen alla tarjottavat kokonaisuudet sijoittuvat kyvykkyyksissään laajalle skaalalle. Vähemmän kyvykkäät sovellukset eivät välttämättä kykene suoriutumaan halutulla tavalla ja parhaat kokonaisratkaisut luontaisesti maksavat paljon. Tehdyn työn monistaminen on virtualisointiratkaisujen ehkä arvokkain potentiaali. Kun datan virtualisointiohjelmiston sisällä tehty moottori on vahva, se voidaan valjastaa moneen eri käyttötarkoitukseen ja muokkaukset sen sisällä voidaan tehdä ketterästi puuttumatta alkupään tai loppupään järjestelmiin.

5.2 Tulevaisuuden näkymät

Haastatteluista ja keskusteluista 2M-IT:n henkilöstön kanssa on käynyt ilmi, että tulevaisuudessa on potentiaalia datan määrän ja käytön kasvulle. Tämä kasvu koostuu lähteiden määrästä, erityyppisen datan hyödyntämismahdollisuuksista ja datan volyymin kasvamisesta. Nämäkin ovat osa niitä globaaleja trendejä, joita Health 4.0 -kirja kuvailee (Thuemmler, 2017). Lisäykset teknologisessa toiminnassa, kuten mobiili-terveyslaitteiden ja IoT-teknologian toiminta yhdessä kehittyvien verkkoyhteyksien kanssa, tuottavat lisää dataa sitä mukaa kun kyseisiä tekniikoita otetaan käyttöön. Jos tarkastellaan tilannetta Suomen näkökulmasta, on havaittavissa ikäjakauman mukainen muutos terveyspalveluiden käyttäjissä. Tämä johtaa siihen, että yhä suurempi osa terveyspalveluiden tarvitsijoista on iäkkäitä ihmisiä. Tilastollisesti iäkkäämmät ihmiset tarvitsevat enemmän ter-

veyspalveluita, jolloin kysyntä luonnollisesti kasvaa näille palveluille. Näistä mahdollisista kehityssuunnista, joita tässä käsitellään, toinen on väistämätön eli väestön ikääntyminen. Toinen eli teknologinen askel eteenpäin ihmisen terveydentilan seurannassa on riippuvainen teknologian kehityksestä ja sen käyttöönoton onnistumisesta. Kumpikin kehityssuunta, eli väestön ikääntyminen ja teknologinen kehitys, kuitenkin johtaa tiedonkäytön lisääntymiseen, mikä merkitsee tiedonjalostuksen ja analysoinnin lisätarvetta ja jos molemmat tapahtuvat samaan aikaan, on vaikutus kumulatiivinen. Vaikka molemmat asettavat haasteita tiedonkäytön kehitykselle, voidaan mahdollisilla ratkaisuilla keventää myös muuta kuormitusta terveydenhuollon systeemistä, sillä saavutetut ratkaisut hyödyttävät kaikkia asiakasryhmiä. Esimerkkinä tästä toimii ennaltaehkäisevä terveystilan seuranta, jolloin kuormittavammilta toimenpiteiltä voidaan mahdollisesti säästyä puuttumalla tilanteeseen aikaisemmin ja tarjoamalla sopivaa hoitoa sopivaan aikaan.

Samaan aikaan kun äsken mainitut demografiset muutokset terveydenhuollossa lisäävät terveyspalveluiden kysyntää, Health 4.0 -kirjassa huomioidaan, että sairaaloiden potilaspaikat ovat samaan aikaan vähentyneet (Thuemmler, 2017). Nykyajan sairaanhoito pyrkii kohti ketterää toimintaa, jossa ylimääräiset sairaalapäivät minimoidaan. Tämä on onnistunut toistaiseksi lääketieteen kehityksen mukana mm. tähystysleikkausten yleistyessä, jolloin leikkaushaavat ovat pienempiä. Laskenut vuodepaikkakapasiteetti saattaa tarkoittaa riskiä tulevaisuuden terveydenhoitokriisistä, jos tarve aktiiviselle hoidolle ylittää terveydenhoitojärjestelmän senhetkisen kapasiteetin. Tietopalveluiden toiminta voi auttaa tässä vaiheessa kahdella tavalla. Perusterveydenhuollon ketteryyden ja potilaan nopean läpimenoajan takaajaksi tarvitaan dynaamista tiedonkulkua ja käsittelyä, jotta jokapäiväinen toiminta pysyy tehokkaana. Lisäksi ennaltaehkäisevän hoidon onnistuminen vaatii toimivaa analyysikoneistoa ja tiedonkeräystä. Näissä ongelmissa datan virtualisoinnin ratkaisut voivat olla avainasemassa.

LÄHTEET

Andreu-Perez, J. *ym.* (2015) ”Big Data for Health”, *IEEE Journal of Biomedical and Health Informatics*, 19(4), ss. 1193–1208. doi: 10.1109/JBHI.2015.2450362.

Brooks (1987) ”No Silver Bullet Essence and Accidents of Software Engineering”, *Computer*, 20(4), ss. 10–19. doi: 10.1109/MC.1987.1663532.

Buchholz, W. (1962a) *Planning a Computer System: Project Stretch*. McGraw-Hill.

Buchholz, W. (1962b) *Planning a Computer System: Project Stretch - International Business Machines Corporation - Google-kirjat*. McGraw-Hill. Saatavissa: https://books.google.fi/books/about/Planning_a_Computer_System.html?id=VcQmAAAAMAAJ&redir_esc=y (Viitattu: 30. huhtikuuta 2019).

Chin, A. G. (2001) *Text databases and document management : theory and practice*. Idea Group Pub. Saatavissa: https://tut.finna.fi/Record/tutcat.149956 (Viitattu: 11. syyskuuta 2018).

Codd, E. F. (1969) ”Derivability, redundancy and consistency of relations stored in large data banks”, *IBM Research report*, RJ 599 (#12343), s. 15. doi: 10.1145/1558334.1558336.

Date, C. J. (1977) *An introduction to database systems*. Addison-Wesley Pub. Co. Saatavissa: https://tut.finna.fi/Record/tutcat.40232 (Viitattu: 11. syyskuuta 2018).

Date, C. J. (1995) *An introduction to database systems*. Addison-Wesley Pub. Co (The systems programming series). Saatavissa: https://tut.finna.fi/Record/tutcat.108666 (Viitattu: 11. syyskuuta 2018).

Desai, B. C. (1990) *An introduction to database systems*. West Pub. Co. Saatavissa: https://tut.finna.fi/Record/tutcat.41322 (Viitattu: 11. syyskuuta 2018).

Google Trends (2019). Saatavissa: https://trends.google.fi/trends (Viitattu: 11. huhtikuuta 2019).

Hovi, A. (1997) *Data warehousing : tietovarastotekniikka*. Suomen atk-kustannus. Saatavissa: https://tut.finna.fi/Record/tutcat.122572 (Viitattu: 11. syyskuuta 2018).

Infographic: The Four V’s of Big Data | IBM Big Data; Analytics Hub (2018) *IBM*. Saatavissa: https://www.ibmbigdatahub.com/infographic/four-vs-big-data (Viitattu: 12. maaliskuuta 2019).

Inmon, W. H. (1998) *Data warehouse performance*. John Wiley. Saatavissa: https://tut.finna.fi/Record/tutcat.131097 (Viitattu: 11. syyskuuta 2018).

Jancsurak, J. (2013) *Watson, knowledge, wisdom ... and transitions - ABI/INFORM Collection - ProQuest*, *Medical Design News*. Saatavissa: https://search-proquest-com.libproxy.tut.fi/abicomplete/docview/1771887082/abstract/C2DE4B8A9F304BB7P

Q/1?accountid=27303 (Viitattu: 30. lokakuuta 2018).

Khine, P. P. ja Wang, Z. S. (2018) ”Data lake: a new ideology in big data era”, *ITM Web of Conferences*. Toimittanut K. Eguchi ja T. Chen, 17, s. 03025. doi: 10.1051/itmconf/20181703025.

van der Lans, R. (2010) *Clearly Defining Data Virtualization, Data Federation, and Data Integration by Rick van der Lans - BeyeNETWORK*, *BeyeNetwork.com*. Saatavissa: http://www.b-eye-network.com/view/14815 (Viitattu: 11. maaliskuuta 2019).

van der Lans, R. (2012) ”Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses”. Morgan Kaufmann, Elsevier. Saatavissa: https://books.google.pt/books?id=7eMd-46wXdMC.

Linstedt, D. *ym.* (2016) ”Introduction to Data Warehousing”, *Data Vault 2.0*. Morgan Kaufmann, ss. 1–15. doi: 10.1016/B978-0-12-802510-9.00001-5.

McFadden, F. R., Hoffer, J. A. ja Prescott, M. B. (1999) *Modern database management.* 5th ed. Addison-Wesley. Saatavissa: https://tut.finna.fi/Record/tutcat.132399 (Viitattu: 11. syyskuuta 2018).

Sundgren, B. (1975) *Theory of data bases*. Petrocelli/Charter. Saatavissa: https://tut.finna.fi/Record/tutcat.48867 (Viitattu: 11. syyskuuta 2018).

Teorey, T. J. ja Fry, J. P. (1982) *Design of database structures*. Prentice-Hall. Saatavissa: https://tut.finna.fi/Record/tutcat.39186 (Viitattu: 11. syyskuuta 2018).

Thuemmler, C. (2017) *How Virtualization and Big Data are Revolutionizing Health Care*. Springer International Publishing. doi: 10.1007/978-3-319-47617-9.

Ullman, J. D. (1988) *Principles of database and knowledge-base systems*. Computer Science Press. Saatavissa: https://tut.finna.fi/Record/tutcat.86678 (Viitattu: 11. syyskuuta 2018).

Ullman, J. D. ja Widom, J. (1997) *A first course in database systems*. Prentice Hall. Saatavissa: https://tut.finna.fi/Record/tutcat.124258 (Viitattu: 11. syyskuuta 2018).

Yuhanna, N., Leganza, G. ja Perdoni, R. (2017) *The Forrester Wave™: Enterprise Data Virtualization, Q4 2017*. Saatavissa: https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Virtualization+Q4+2017/-/E-RES133042# (Viitattu: 21. elokuuta 2018).

Yuhanna, N., Owens, L. ja Cullen E. (2015) *The Forrester Wave™: Enterprise Data Virtualization, Q1 2015*. Saatavissa: https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Virtualization+Q1+2015/-/E-RES117844 (Viitattu: 21. elokuuta 2018).

Zaidi, E., Beyer, M. ja Jain, A. (2017) *Market Guide for Data Virtualization*, *Gartner*. Saatavissa: https://www.gartner.com/doc/3778873/market-guide-data-virtualization (Viitattu: 16. lokakuuta 2018).

LIITE A: ASIANTUNTIJAHAASTATTELUIDEN RUNKO

1. Tausta, asiantuntijat

Millainen koulutus?

Mikä on oma osaamisala?

Miksi tämä ala?

Mitä käytännön työssä tehdään?

Minkälaisia projekteja?

2. Nykyhetki, tarpeet ja tavoitteet

Kuka on asiakas ja kuka on loppukäyttäjä?

Millaisia ovat loppukäyttäjien tarpeet?

Millaisia ovat asiakkaiden tarpeet?

Mistä tietää vastataanko tarpeisiin?

Ohjelmistotyö ja asiakassuhde

Projektityön luonne

3. Nykyhetki, tekniikat

Millaisia ovat yleiset ongelmat, joita tekniikalla ratkaistaan? Onko tekniikassa ongelmia?

Mikä on isoin ongelma yleisesti? Tekniikassa? Vaikein?

Onko ongelmia mitä ei kyetä ratkaisemaan?

Missä kuluu eniten työaikaa?

Millä työkaluilla ja menetelmillä työtä tehdään?

Mikä toimii hyvin tekniikassa?

Onko vaihtoehtoisia työkaluja tai menetelmiä? Etuja? Haittoja?

Data, mistä tulee, mihin menee?

Näkemys datan virtualisoinnista nykytekniikkana?

4. Tulevaisuus, näkymät

Miten tarpeet kehittyvät tulevaisuudessa asiakkailla?

Miten ongelmat kehittyvät tulevaisuudessa?

Määrä vs moninaisuus? Lähteet?

Miten oma työväline tai tekniikka jatkuu tulevaisuudessa?

Miten tekniikka kehittyy?

Miten oma työ kehittyy ja miten ala kehittyy?